

**ANALISIS DAN IMPLEMENTASI INTERKONEKSI IPv6
DAN IPv4 DENGAN APLIKASI TUNNEL BROKER
DI LABORATORIUM JARINGAN KOMPUTER**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Teknik pada
Jurusan Teknik Elektro



Oleh :

CANDRA GUNAWAN

10655004518

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU**

2013

ANALISIS DAN IMPLEMENTASI INTERKONEKSI IPv6 DAN IPv4 DENGAN APLIKASI TUNNEL BROKER DI LABORATORIUM JARINGAN KOMPUTER

CANDRA GUNAWAN
NIM : 10655004518

Tanggal Sidang : 27 Juni 2013
Tanggal Wisuda : November 2013

Jurusan Teknik Elektro
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. Soebrantas No. 155 Pekanbaru

ABSTRAK

Pesatnya kemajuan teknologi di bidang jaringan komputer membuat alokasi alamat Internet Protokol versi 4 (IPv4) semakin menipis. Untuk mengatasi kelangkaan alokasi alamat ini dibutuhkan sistem pengalokasian alamat yang baru dan memiliki *range* alamat yang besar maka dipilihlah Internet Protokol versi 6 (IPv6). Kegunaan dan manfaat IPv6 ini masih belum dirasakan bagi kebanyakan pengguna internet saat ini, namun bagi beberapa *webmaster* dan *user* yang bereksperimen dengan IPv6 kegunaan implementasi ini sungguh sangat berguna dalam menggali pengetahuan dan mengimplementasikan IPv6 walaupun IPv6 ini masih dalam tahap transisi, sehingga *website* dan aktifitas IPv6 yang dibuat masih bertahan sampai alokasi alamat IPv4 benar-benar habis. Pada penelitian ini digunakan *tunnel broker* dengan *software Tunnel Setting Protocol* (TSP) yang dikonfigurasi secara otomatis sehingga implementasi menjadi semakin ringkas untuk mengoperasikan IPv6 yang terenkapsulasi IPv4. untuk mencari perbandingan *delay* rata-rata transfer data antara *tunnel broker* dengan IPv4 dilakukan analisis dengan wireshark. Percobaan seperti *download* dan *upload*, maka hasil yang didapat pada tiap-tiap percobaan akan dibandingkan antara IPv4 dengan *tunnel broker* dan didapat hasil bahwa IPv6 lebih cepat dalam proses *upload* dibanding IPv4 karena paket pelengkap yang bukan termasuk data yang ditransfer pada IPv6 lebih sedikit dibanding IPv4 sehingga waktu yang diproses pun lebih sedikit.

Kata Kunci : *delay, IPv4, Tunnel Broker, Tunnel Setting Protocol (TSP), Wireshark*

**ANALYSIS AND IMPLEMENTATION OF INTERCONNECTION IPv6
AND IPv4 WITH APPLICATIONS TUNNEL BROKER
IN THE COMPUTER NETWORK LABORATORY**

**CANDRA GUNAWAN
NIM : 10655004518**

*Date of Final Exam : June 27 2013
Graduation date: November 2013*

*Department of Electrical Engineering
Faculty of Science and Technology
Islamic State University of Sultan Syarif Kasim Riau
Soebrantas St. No. 155 Pekanbaru*

ABSTRACT

The rapid technological advances in the field of computer networks make the allocation of Internet Protocol version 4 addresses (IPv4) are running low. To overcome the scarcity of address allocation is needed that addresses the new allocation system and has a huge range of addresses that the chosen Internet Protocol version 6 (IPv6). Uses and benefits of IPv6 is still not perceived to most Internet users today, but for some webmasters and users are experimenting with IPv6 uses this implementation is extremely useful in exploring knowledge and implement IPv6 IPv6 although this is still in the transition phase, so that the website and activities IPv6 made still survive until the allocation of IPv4 addresses completely exhausted. In this study, use tunnel broker with the Tunnel software Setting Protocol (TSP) which is automatically configured so that implementation becomes increasingly quick to operate IPv6 encapsulated IPv4. to search for comparison of the average delay of data transfer between IPv4 tunnel broker to do the analysis with Wireshark. Experiments such as download and upload, the results obtained in each experiment will be compared between the tunnel broker IPv4 and IPv6 obtained results that much faster than the upload process of IPv4 as a complementary package that is not included in the data being transferred is less than the IPv4 IPv6 so processing time was less.

Keywords: *delay, IPv4, Tunnel Broker, Tunnel Setting Protocol (TSP), Wireshark*

KATA PENGANTAR

Assalamualaikum, Wr, Wb

Puji syukur penulis ucapkan kepada Allah SWT Tuhan semesta alam yang telah melimpahkan rahmat dan hidayahnya sehingga penulis dapat menyelesaikan tugas akhir ini. Salawat beserta salam kita panjatkan kepada junjungan alam yakni Nabi besar kita Muhammad SAW. Dengan limpahan kasih sayang Allah SWT penulis dapat menyelesaikan penelitian tugas akhir yang berjudul “Analisis dan Implementasi Interkoneksi IPv6 dan IPv4 dengan Aplikasi *Tunnel Broker* di Laboratorium Jaringan Komputer”. Tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan studi S1 di jurusan Teknik Elektro, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.

Pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang membantu penulis baik itu berupa moral, materil, ataupun berupa pikiran sehingga terlaksananya penelitian dan penulisan laporan ini, antara lain kepada :

1. Kedua orang tua tercinta, yang sangat penulis sayangi dan seluruh anggota keluarga atas segala do'a, nasihat dan kasih sayang yang tidak terhingga besarnya.
2. Ibu Dra. Hj. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi.
3. Bapak Kunaifi, ST, PgDipEnSt, M.Sc, selaku Ketua Jurusan Teknik Elektro Fakultas Sains dan Teknologi UIN Suska Riau.
4. Bapak Abdillah, SSi., MIT selaku Dosen Pembimbing Tugas Akhir.
5. Bapak Dr. Alex Wenda, ST., Meng, selaku Dosen Penguji I.
6. Ibu Ewi Ismaredah, S. Kom., M. Kom, selaku Dosen Penguji II.
7. Seluruh Dosen Jurusan Teknik Elektro Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau yang telah banyak membimbing.
8. *My brothers* (Didi Yuda Prawira, Imam Wahyudi) *and my sister* (Amaliyah Pratiwi),
9. Teman seperjuangan angkatan 2006 :
Ilham Wahid (Bakteri), Defteriandi Muttakin (DR.M), Rio nursan (Slash Mokondo), Deny Prayuda Atmayewa (Pakde), Budi Kurnia Syahputra (Kimbud), Sandrio Irwan (Kimbek), Febri Erizal (Ijak), Willy Irawan (Pak Will), Ucok (Aseng), harminsyah (Bung Harmin), Agus Sumardi (ocu), Arifin (Si Lai), Feri Syahputra, Adi (Gapuak), Yudi

(Ambon), Ades, Siti Habibah, Dian, Mardha (Upiak), Ahmad Faizal (Ocu ical), Khairul Fahri (Ocu Irul), Toga Agung P. (toge) dan lain - lain.

10. Senior dan Junior Teknik Elektro UIN SUSKA RIAU.
11. Fitra Wardhana yang memberi sedikit pencerahan yang berarti untuk penyelesaian Tugas Akhir ini.
12. Semua pihak yang telah membantu penulis dalam mengerjakan laporan ini yang tidak dapat penulis sebutkan satu persatu.

Laporan penelitian ini merupakan salah satu syarat kelulusan untuk menyelesaikan studi S1 Jurusan Teknik Elektro, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.

Penulis sangat menyadari bahwa penelitian ini belum sempurna adanya, sehingga kritik dan saran dari seluruh pembaca sangat penulis harapkan demi kesempurnanya laporan penelitian ini. Semoga Allah SWT, Yang Maha Pengasih lagi Maha Penyayang, melimpahkan rahmat-Nya kepada Bapak/Ibu serta rekan-rekan, sebagai imbalan atas segala jasa yang telah diberikan kepada penulis. Demikian pula semoga penelitian ini dapat memberikan manfaat kepada kita semua umumnya. Khususnya bagi teman-teman yang menekuni ilmu yang sama.

Pekanbaru, 27 Juni 2013

Penulis

DAFTAR ISI

	Hal
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR RUMUS	xvi
DAFTAR SINGKATAN	xvii
DAFTAR LAMPIRAN	xviii
BAB I PENDAHULUAN	
1.1. Latar Belakang	I-1
1.2. Rumusan Masalah	I-2
1.3. Tujuan Penelitian.....	I-2
1.4. Batasan Masalah.....	I-2
1.5. Manfaat Penelitian	I-2
BAB II TINJAUAN PUTAKA	
2.1. Penelitian Terkait	II-1
2.2. Teori Jaringan Komputer	II-1
2.3. Teori Internet Protokol	II-2
2.3.1. Internet Protokol versi 4 (IPv4)	II-2
2.3.1.1. Jenis-Jenis Alamat IPv4	II-3

2.3.1.2. Kelas-Kelas Alamat IPv4	II-3
2.3.2. Internet Protokol versi 6 (IPv6)	II-5
2.3.2.1. Struktur IPv6	II-6
2.3.2.2. Format Alamat IPv6	II-6
2.3.2.3. Format <i>Prefix</i> IPv6	II-7
2.4. Mekanisme Interkoneksi.....	II-7
2.4.1. <i>DualStack</i>	II-8
2.4.2. <i>Tunneling</i>	II-8
2.5. <i>IPv6 Tunnel Broker</i>	II-8
2.5.1. <i>6in4 Tunnel Broker</i>	II-9
2.5.2. <i>4in6 Tunnel Broker</i>	II-10
2.6. <i>Gogo6 Tunnel Broker</i>	II-10
2.7. <i>Transfer Control protocol (TCP)</i>	II-12
2.8. <i>Monitoring dengan Wireshark</i>	II-14

BAB III METODE PENELITIAN

3.1. Jenis Penelitian	III-1
3.2. Tahapan Penelitian	III-1
3.2.1 Perumusan Masalah	III-2
3.2.2. Pengumpulan Data	III-2
3.2.3. Analisa <i>Tunnel Broker</i>	III-2
3.2.4. Perancangan Interkoneksi IPv6 dan IPv4 dengna Aplikasi <i>Tunnel Broker</i>	III-3
3.2.4.1. Diagram Alir (<i>Flow Chart</i>) Perancangan Interkoneksi IPv6 dan IPv4 ...	III-3
3.2.4.2. Skema Perancangan interkoneksi IPv6 dan IPv4 pada Aplikasi <i>Tunnel Broker</i>	III-4
3.2.5. Analisis Perangkat Jaringan pada Laboratorium	III-4
3.2.6. Implementasi	III-5
3.2.7. Pengujian dan Analisis	III-5

BAB IV IMPLEMENTASI DAN ANALISIS

4.1. Implementasi Sistem	IV-1
4.1.1. Konfigurasi <i>Host</i>	IV-2
4.1.2. konfigurasi <i>Tunnel Broker</i>	IV-3
4.1.3. Pengujian Sistem	IV-6
4.1.3.1. Pengujian dengan <i>Ping</i>	IV-6
4.1.3.2. Pengujian dengan <i>Browser</i>	IV-7
4.2. Analisis Perbandingan <i>Delay</i>	IV-8
4.2.1. Analisis <i>Delay</i> Aktivitas <i>Download</i>	IV-8
4.2.2. Analisis <i>Delay</i> Aktivitas <i>Upload</i>	IV-16
4.2.3. <i>Delay</i> Keseluruhan	IV-24
4.2.4. Analisis Hasil Implementasi	IV-25

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan	V-1
5.2. Saran	V-1

DAFTAR PUSTAKA

LAMPIRAN A

LAMPIRAN B

DAFTAR RIWAYAT HIDUP

BAB I

PENDAHULUAN

1.1. Latar Belakang

Internet Protocol (IP) adalah metode untuk mengirimkan data ke internet. Setiap komputer dalam internet setidaknya harus mempunyai sebuah alamat IP yang unik yang mengidentifikasikan komputer tersebut terhadap komputer yang lainnya. Dalam jaringan internet ada beberapa jenis jaringan seperti *Local area Network* (LAN) dan *Wide Area Network* (WAN) atau dalam beberapa literatur juga disebut Internet. *Internet Protocol version 4* (IPv4) yang menjadi basis sistem pengalamatan saat ini dapat dengan bebas digunakan untuk jaringan bersekala kecil seperti LAN, namun untuk implementasi pada WAN alokasi alamat yang dimilikinya semakin menipis. Maka diperkenalkan sistem pengalamatan yang baru dan lebih besar alamatnya yaitu *Internet Protocol version 6* (IPv6).

Kegunaan dan manfaat IPv6 ini masih belum dirasakan bagi kebanyakan pengguna internet saat ini, namun bagi beberapa *webmaker* dan *user* yang bereksperimen dengan IPv6 kegunaan implementasi ini sungguh sangat berguna dalam menggali pengetahuan dan mengimplementasikan IPv6 walaupun IPv6 ini masih dalam tahap transisi, sehingga *website* dan aktifitas IPv6 yang dibuat masih bertahan sampai alokasi alamat IPv4 benar-benar habis.

Konektifitas IPv6 dengan IPv4 memiliki sistematika yang berbeda dan untuk menyatukan keduanya dibutuhkan sebuah mekanisme interkoneksi seperti tunneling. Namun dalam *tunneling* ini sendiri memiliki waktu transfer data yang berbeda dengan sistem pengalamatan IPv4 yang ada saat ini, dari perbedaan ini peneliti mencoba untuk menganalisis dan membandingkan keduanya hingga didapat hasil yang signifikan dan akurat.

Untuk analisis kinerja IPv6 ini penulis menggunakan *software Wireshark*. *Wireshark* adalah salah satu dari sekian banyak *tool Network Analyzer* yang banyak digunakan oleh *Network administrator* untuk menganalisa kinerja jaringannya. *Wireshark* lebih disukai oleh para *administrator* jaringan karena *interface*-nya menggunakan *Graphical User Interface* (GUI) atau tampilan grafis dan semua jenis paket informasi dalam berbagai format protokol pun akan dengan mudah ditangkap dan dianalisa.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas maka penulis mengambil rumusan masalah sebagai berikut :

1. Bagaimana mengimplementasikan konektifitas IPv6 dimana *provider* Internet menggunakan IPv4
2. Bagaimana menganalisis *delay* ketika paket-paket data IPv6 dikirim maupun diterima *host* ke internet dan membandingkan waktunya dengan IPv4

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian atas Tugas akhir ini adalah Analisis dan Implementasi Interkoneksi IPv6 dan IPv4 pada Aplikasi *Tunnel Broker* di Laboratorium Jaringan Komputer.

1.4. Batasan Masalah

Dalam penelitian Tugas Akhir dengan judul *Analisis dan Implementasi Interkoneksi IPv6 dan IPv4 pada Aplikasi Tunnel Broker di Laboratorium Jaringan Komputer* dibatasi oleh beberapa hal berikut:

1. Analisis yang dilakukan untuk membandingkan antara *Tunneling* IPv6 dan IPv4 adalah analisis *delay* yang didapat saat proses *download* dan *upload*.
2. *Software* digunakan untuk *Tunnel Broker* adalah *gogo6* dan GUI *gogoClient* sebagai aplikasi pada *host* dan program analisis yang digunakan adalah *Wireshark*.

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini dapat diaplikasikan pada Laboratorium Jaringan Komputer Fakultas Sains dan Teknologi UIN Suska Riau antara lain :

1. Konektifitas IPv6 dapat diimplementasikan sehingga dapat digunakan bagi sebagian *web master* untuk membangun *website* berbasis IPv6.
2. Komputer di Laboratorium yang pada dasarnya menggunakan IPv4 bisa terkoneksi dengan internet yang menggunakan pengalamatan IPv6.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Penelitian mengenai Analisis dan Implementasi IPv6 pada jaringan IPv4 dengan *Tunnel broker* di Laboraturium jaringan komputer sendiri sebenarnya telah dilakukan oleh beberapa penelitian di Indonesia. Beberapa penelitian tersebut antara lain :

1. Penelitian Mochammad Syarif Averoes (2012) yang berjudul *Analisa Unjuk Kerja Aplikasi Video Streaming Pada jaringan IPv6-Dual Stack dengan menggunakan PC Router & Emulator GNS3*. Penelitian ini bertujuan merancang dan membangun suatu aplikasi video *Streaming* pada jaringan IPv6 *Dual Stack* dan dianalisa dengan PC router dan sebagai pembandingnya adalah *Emulator* GNS3.
2. Penelitian Reko Artondo (2010) yang berjudul *Analisa dan Implementasi IPv6 Tunnel Broker untuk Interkoneksi Antara IPv6 dengan IPv4*. Penelitian ini mencoba membangun sebuah *Tunnel Server* untuk dapat mengimplementasikan *Tunnel Broker* yang dapat mengenkapsulasi data IPv6 kedalam IPv4 dan menganalisa menggunakan Wireshark.
3. Penelitian Renny Indah Lestari (2011) dengan judul *Menganalisa Kinerja Antara Metode Tunneling 6t04 dengan Metode Dual Stack Berbasis Protokol IPv6 Menggunakan Router Mikrotik (Studi kasus PT. Time Excelindo)*. Penelitian ini menganalisa dua buah metode transisi IPv4 ke IPv6 dengan aplikasi *Tunnel Broker*.

Sebagai orisinalitas dari penelitian sebelumnya yaitu pada Tugas Akhir ini yang berjudul *Analisis Dan Implementasi Interkoneksi IPv6 dan IPv4 dengan Aplikasi Tunnel Broker di Laboraturium Jaringan Komputer* adalah pada implementasi IPv6 yang dienkapsulasi ke dalam paket IPv4 dengan *Tunnel Broker* pada vendor Gogo6/Freenet6 dan *Graphic User Interface* menggunakan gogoCLIENT

2.2. Teori Jaringan Komputer

Jaringan Komputer adalah kumpulan beberapa komputer yang tergabung dalam suatu lingkungan yang dapat saling berkomunikasi satu dengan yang lain. Berdasarkan

media yang digunakan, maka terdapat dua macam jaringan yaitu jaringan dengan kabel dan jaringan nirkabel. Secara umum, jaringan dibagi menjadi 3 jenis :

1. *Local Area Network (LAN)*

LAN merupakan tipe jaringan dengan kecepatan yang tinggi yang meliputi area seperti satu gedung. Tingkat kesalahan dalam pengiriman data rendah karena hanya dalam area yang kecil.

2. *Wide Area Network (WAN)*

WAN merupakan jaringan komunikasi data yang melayani pengguna dalam wilayah area geografi yang luas dan menggunakan peralatan transmisi.

3. *Metropolitan Area Network (MAN)*

MAN merupakan jaringan yang melayani area metropolitan, biasanya area yang ada lebih besar dari LAN dan lebih kecil dari WAN.

2.3 Teori Internet Protokol

2.3.1. Internet Protokol versi 4 (IPv4)

IPv4 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IPv4. Panjang totalnya adalah 32-bit, dan secara teoritis dapat mengalami hingga 4 miliar host komputer atau lebih tepatnya 4.294.967.296 host di seluruh dunia, jumlah host tersebut didapatkan dari 256 didapatkan dari 8 bit dipangkat 4 karena terdapat 4 oktet sehingga nilai maksimal dari alamat IP versi 4 tersebut adalah 255.255.255.255 dimana nilai dihitung dari nol sehingga nilai host yang dapat ditampung adalah $256 \times 256 \times 256 \times 256 = 4.294.967.296$ host, bila host yang ada di seluruh dunia melebihi kuota tersebut maka dibuatlah IP versi 6 atau IPv6.

Alamat IP versi 4 umumnya diekspresikan dalam notasi desimal bertitik yang dibagi ke dalam empat buah oktet berukuran 8-bit. Dalam beberapa buku referensi, format bentuknya adalah w.x.y.z. Karena setiap oktet berukuran 8-bit, maka nilainya berkisar antara 0 hingga 255.

Alamat IP yang dimiliki oleh sebuah *host* dapat dibagi dengan menggunakan subnet mask jaringan ke dalam dua buah bagian, yaitu:

1. *Network Identifier/NetID* atau *Network Address* yang digunakan khusus untuk mengidentifikasikan alamat jaringan di mana host berada. Semua sistem di dalam

sebuah jaringan fisik yang sama harus memiliki alamat *network identifier* yang sama. *Network identifier* juga harus bersifat unik dalam sebuah *Internetwork*. Jika semua *node* di dalam jaringan logis yang sama tidak dikonfigurasi dengan menggunakan *network identifier* yang sama, maka terjadilah masalah yang disebut dengan *routing error*. Alamat *network identifier* tidak boleh bernilai 0 atau 255.

1. *Host Identifier/HostID* atau *Host address* yang digunakan khusus untuk mengidentifikasi alamat host (dapat berupa *workstation*, *server* atau sistem lainnya yang berbasis teknologi TCP/IP) di dalam jaringan. Nilai *host identifier* tidak boleh bernilai 0 atau 255 dan harus bersifat unik di dalam *network identifier*/segmen jaringan di mana ia berada.

2.3.1.1 Jenis-Jenis Alamat IPv4

Alamat IPv4 terbagi menjadi beberapa jenis, yakni sebagai berikut:

1. Alamat *Unicast* merupakan alamat IPv4 yang ditentukan untuk sebuah antarmuka jaringan yang dihubungkan ke sebuah *Internetwork* IP. Alamat *unicast* digunakan dalam komunikasi *point-to-point* atau *one-to-one*.
2. Alamat *Broadcast* merupakan alamat IPv4 yang didesain agar diproses oleh setiap *node* IP dalam segmen jaringan yang sama. Alamat *broadcast* digunakan dalam komunikasi *one-to-everyone*.
3. Alamat *Multicast* merupakan alamat IPv4 yang didesain agar diproses oleh satu atau beberapa *node* dalam segmen jaringan yang sama atau berbeda. Alamat *multicast* digunakan dalam komunikasi *one-to-many*.

2.3.1.2 Kelas-Kelas Alamat IPv4

Dalam RFC 791, alamat IP versi 4 dibagi ke dalam beberapa kelas, dilihat dari oktet pertamanya, seperti terlihat pada tabel 2.1. RFC sendiri adalah *Request for Comments* yaitu salah satu dari seri dokumen informasi dan standar Internet bernomor yang diikuti secara luas oleh perangkat lunak untuk digunakan dalam jaringan, Internet dan beberapa sistem operasi jaringan, mulai dari Unix, Windows, dan Novell NetWare. RFC kini diterbitkan di bawah arahan *Internet Society* (ISOC) dan badan-badan penyusun standar teknisnya, seperti *Internet Engineering Task Force* (IETF) atau *Internet Research Task Force* (IRTF). Semua standar Internet dan juga TCP/IP selalu dipublikasikan dalam RFC.

Sebenarnya yang menjadi pembeda kelas IP versi 4 adalah pola biner yang terdapat dalam oktet pertama, utamanya adalah bit-bit awal/*high-order bit*, tapi untuk lebih mudah mengingatnya, akan lebih cepat diingat dengan menggunakan representasi desimal.

Tabel 2.1 Kelas IPv4

Kelas Alamat IP	Oktet pertama (desimal)	Oktet pertama (biner)	Digunakan oleh
Kelas A	1–126	0xxx xxxx	Alamat <i>unicast</i> untuk jaringan skala besar
Kelas B	128–191	10xx xxxx	Alamat <i>unicast</i> untuk jaringan skala menengah hingga skala besar
Kelas C	192–223	110x xxxx	Alamat <i>unicast</i> untuk jaringan skala kecil
Kelas D	224–239	1110 xxxx	Alamat <i>multicast</i> (bukan alamat <i>unicast</i>)
Kelas E	240–255	1111 xxxx	Direservasikan; umumnya digunakan sebagai alamat percobaan (eksperimen); (bukan alamat <i>unicast</i>)

(Sumber: id.wikipedia.org)

a. Kelas A

Alamat-alamat kelas A diberikan untuk jaringan skala besar. Nomor urut bit tertinggi di dalam alamat IP kelas A selalu diset dengan nilai 0. Tujuh bit berikutnya untuk melengkapi oktet pertama akan membuat sebuah *network identifier*. 24-bit sisanya merepresentasikan *host identifier*. Ini mengizinkan kelas A memiliki hingga 126 jaringan, dan 16,777,214 host tiap jaringannya. Alamat dengan oktet awal 127 tidak diizinkan, karena digunakan untuk mekanisme *Interprocess Communication* (IPC) di dalam mesin yang bersangkutan.

b. Kelas B

Alamat-alamat kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua bit pertama di dalam oktet pertama alamat IP kelas B selalu diset ke bilangan biner 10. 14 bit berikutnya (untuk melengkapi dua oktet pertama), akan membuat sebuah *network identifier*. 16 bit sisanya merepresentasikan *host identifier*. Kelas B dapat memiliki 16,384 network, dan 65,534 host untuk setiap *network*-nya.

c. Kelas C

Alamat IP kelas C digunakan untuk jaringan berskala kecil. Tiga bit pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai biner 110. 21-bit selanjutnya akan membentuk sebuah *network identifier*. 8 bit sisanya sebagai oktet terakhir akan

merepresentasikan *host identifier*. Ini memungkinkan pembuatan total 2,097,152 buah *network*, dan 254 host untuk setiap *network*-nya.

d. Kelas D

Alamat IP kelas D disediakan hanya untuk alamat-alamat *IP multicast*, sehingga berbeda dengan tiga kelas di atas. Empat *bit* pertama di dalam IP kelas D selalu diset ke bilangan biner 1110. 28-*bit* sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali host. Untuk lebih jelas mengenal alamat ini, lihat pada bagian Alamat *Multicast IPv4*.

e. Kelas E

Alamat IP kelas E disediakan sebagai alamat yang bersifat eksperimental atau percobaan dan dicadangkan untuk digunakan pada masa depan. Empat bit pertama selalu diset kepada bilangan biner 1111. 28-bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali host.

Penggunaan kelas alamat IP sekarang tidak relevan lagi, mengingat sekarang alamat IP sudah tidak menggunakan kelas alamat lagi. Pengembangan otoritas Internet telah melihat dengan jelas bahwa alamat yang dibagi ke dalam kelas-kelas seperti di atas sudah tidak mencukupi kebutuhan yang ada saat ini, di saat penggunaan Internet yang semakin meluas. Alamat IPv6 yang baru sekarang tidak menggunakan kelas-kelas seperti alamat IPv4. Alamat yang dibuat tanpa mempedulikan kelas disebut juga dengan *classless address*. (Sumber: id.wikipedia.org)

2.3.2. Internet Protokol versi 6 (IPv6)

Perkembangan internet yang sangat pesat saat ini menyebabkan alokasi alamat IPv4 semakin berkurang, hal ini menyebabkan harga IP address legal sangat mahal. Untuk mengatasi kekurangan alokasi IP address maka IETF mendesain suatu IP baru yang disebut *Internet Protocol versi 6 (IPv6)*.

IPv6 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan Protokol Internet versi 6. Panjang totalnya adalah 128-bit, dan secara teoritis dapat mengalami hingga $2^{128} = 3,4 \times 10^{38}$ *host* komputer di seluruh dunia.

2.3.2.1. Struktur IPv6

Pada tahun 1992 IETF selaku komunitas terbuka internet membuka diskusi para pakar untuk mengatasi masalah ini dengan mencari format alamat IP generasi berikutnya setelah IPv4 atau *IP next generation (IPng)* yang kemudian menghasilkan banyak *Request For Comments (RFC)* yakni dokumen standard yang membahas protokol, program, prosedur serta konsep internet IPv6.

Setelah melalui pembahasan yang panjang, pada tahun 1995 ditetapkan melalui RFC:2460 alamat IP versi 6 sebagai IPng pengganti IP versi 4.

IPv6 ini menggunakan format 128-bit sehingga bisa menampung kebutuhan:

128-bit

= 2^{128} IPv6 Address

= 340,282,366,920,938,463,374,607,431,768,211,456 IPv6 Address

Pengembangan IPv6 sampai saat ini sudah dilakukan oleh banyak pihak yang ada di seluruh dunia termasuk *Service Provider, Internet Exchange Point, ISP regional, Militer serta Universitas*.

2.3.2.2. Format Alamat IPv6

Terdapat tiga cara umum untuk merepresentasikan alamat IPv6 sebagai teks yaitu:

- a. Bentuk yang diacu adalah x:x:x:x:x:x:x dimana x adalah nilai heksadesimal dari kedelapan 16 bit dari alamat. Contoh:

1) FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

2) 1080:0:0:0:8:800:200C:417A

- b. Karena dimungkinkan alamat yang memiliki rentetan nilai 0 yang panjang, maka untuk menyederhanakannya dapat digunakan bentuk penulisan khusus untuk memadatkan 0 tersebut. Bentuk penulisan tersebut adalah :: yang menandakan ada lebih dari satu grup yang mengandung 0. Tanda :: hanya boleh muncul satu kali dalam alamat. Tanda :: dapat pula digunakan untuk memadatkan *leading* atau *trailing* 0 pada alamat. Contoh:

- 1) Alamat 1080:0:0:0:8:800:200C:417A dapat direpresentasikan sebagai :

1080::8:800:200C:417A

- 2) Alamat 0:0:0:0:0:0:0:1 dapat direpresentasikan sebagai ::1

c. Bentuk lain adalah $x:x:x:x:x:d.d.d$, dimana x adalah nilai heksadesimal dari keenam 16 bit bagian urutan teratas dari alamat dan d adalah nilai desimal dari keempat 8 bit urutan terendah dari alamat. Contoh:

1) 0:0:0:0:0:0:13.1.68.3 dimana ini dapat disingkat menjadi ::13.1.68.3

2) 0:0:0:0:0:FFFF:129.144.52.38 dan dapat disingkat menjadi ::FFFF:129.144.52.38

2.3.2.3. Format *Prefix* IPv6

Terdapat tiga jenis alamat yaitu:

a. Unicast

Alamat ini digunakan untuk kartu jaringan tunggal. Sebuah paket yang dikirim ke alamat ini akan diberikan ke kartu jaringan yang memiliki alamat ini.

b. Anycast

Alamat ini diperuntukkan bagi himpunan kartu jaringan yang biasanya dimiliki oleh *host-host* yang berbeda. Suatu paket yang dikirim ke alamat ini akan diberikan ke salah satu kartu jaringan yang dikenali oleh alamat ini. Biasanya kartu jaringan yang dipilih adalah yang terdekat menurut ukuran protokol *routing*.

c. Multicast

Alamat ini diperuntukkan bagi himpunan kartu jaringan yang biasanya dimiliki oleh *host-host* yang berbeda. Suatu paket yang dikirim ke alamat ini akan diberikan ke seluruh kartu jaringan yang dikenali oleh alamat ini. Dengan demikian alamat *broadcast* pada IPv4 telah digantikan dengan jenis alamat ini. (Sumber: Mohamad Eko, 2009)

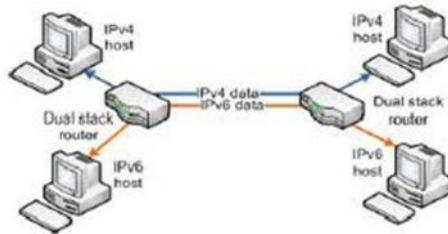
2.4. Mekanisme Interkoneksi

Secara garis besar implementasi IPv6 tidak dapat secara langsung dilakukan di semua lini *end-to-end*, terkait dengan keterlibatan jumlah komunitas/organisasi yang sangat besar di Internet, banyaknya aplikasi berbasis IPv4 yang telah digunakan, dan banyaknya bisnis yang masih memanfaatkan IPv4. Hal yang akan terjadi adalah adanya fase transisi secara bertahap dari IPv4 ke IPv6 dan implementasi IPv6 dengan IPv4 selama renggang waktu yang tidak dapat diprediksi.

Namun demikian desain IPv6 sudah menyertakan mekanisme transisi. Beberapa mekanisme transisi tersebut yaitu:

2.4.1. DualStack

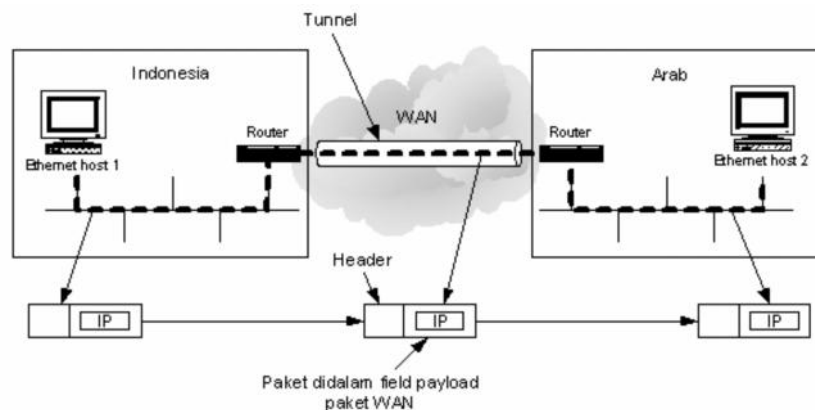
DualStack Merupakan sebuah mekanisme transisi IPv4 dengan IPv6 yang menggunakan dua buah tipe IP pada sebuah *node*, *DualStack* diimplementasikan pada *network layer*, sehingga pada lapis ini terdapat dua jenis IP yang berbeda.



Gambar 2.1 *DualStack*

2.4.2. Tunneling

Tunneling menggunakan IPv4 sebagai media penghubung antar jaringan IPv6. Paket IPv6 dibungkus dengan paket IPv4, sehingga paket IPv6 pada jaringan IPv4 diperlakukan sebagai *payload* biasa. Pada *end point*, *header* IPv4 dibuka, sehingga yang diperoleh adalah paket IPv6. Contoh mekanisme *tunneling* ini *Tunnel broker*, *6to4*, *6over4*, *automatic tunnel*.

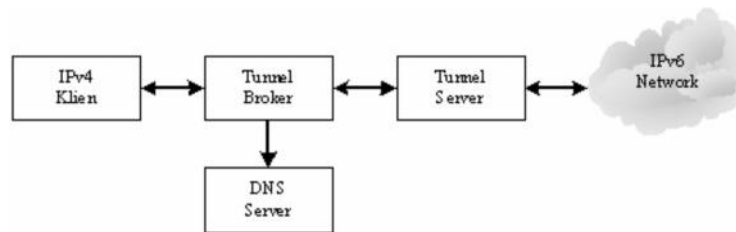


Gambar 2.2 *Tunneling* suatu paket dari Indonesia ke Arab
(Sumber: Reko, 2010)

2.5 IPv6 Tunnel Broker

IPv6 tunnel broker merupakan sub metode dari mekanisme interkoneksi *tunneling*. Dalam konteks jaringan komputer, sebuah *tunnel broker* adalah *software* yang menyediakan jaringan terowongan (*tunnel*). *Tunnel* ini dapat menyediakan konektivitas dikemas melalui infrastruktur yang ada untuk infrastruktur lain. *Tunnel Broker* termasuk

kedalam Metode *Tunneling*, *Tunnel Broker* juga dapat diartikan sebagai *ISP virtual*, fasilitas ini dapat menjadi alternatif saat sebuah situs tidak bisa diakses karena perbedaan versi pengalaman protokol.



Gambar 2.3 Arsitektur IPv6 *Tunnel Broker*

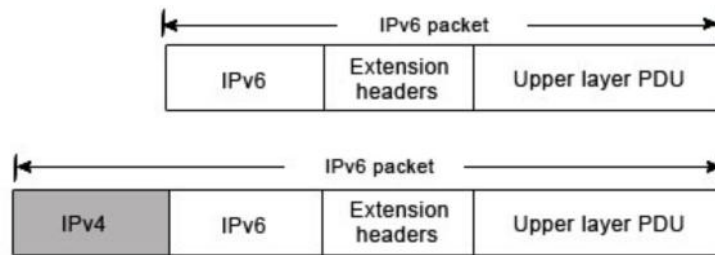
(Sumber: Reko, 2010)

Ada berbagai *Tunnel Broker*, meskipun paling sering istilah yang digunakan untuk merujuk ke *Tunnel Broker* IPv6, sebagaimana didefinisikan dalam RFC: 3053, tetapi juga dapat merujuk ke *tunnel Broker* IPv4. *Tunnel Broker* IPv6 umumnya menyediakan terowongan IPv6 ke situs atau pengguna akhir IPv4. Dalam *Tunnel broker* umum menawarkan apa yang disebut protokol 41 atau *41-Tunnel Protocol*. Ini adalah terowongan di mana *Tunnel IPv6* langsung di dalam IPv4 paket dengan memiliki satu protokol lapangan diatur ke 41 (IPv6) di paket IPv4. Dalam kasus IPv4 jalur *Tunnel broker* IPv4 disediakan untuk pengguna dengan enkapsulasi IPv4 dalam IPv6 sebagaimana didefinisikan dalam RFC: 2473.

Konfigurasi tunnel IPv6 biasanya dilakukan dengan menggunakan *Tunnel Setting Protocol* (TSP), atau menggunakan protokol *Tunnel Informasi Control* (TIC). Pada klien menggunakan *Automatic IPv6 Connectivity Utilitas Client* (AICCU). Selain IPv6, tunnel TSP juga dapat digunakan untuk mengatur *tunnel* IPv4. Ada dua jenis *Tunnel Broker* yang disediakan untuk berbagai keperluan yaitu:

2.5.1. 6in4 Tunnel Broker

Tunnel broker jenis ini adalah layanan penyedia jaringan enkapsulasi IPv6 kedalam IPv4. Cara ini digunakan oleh pengguna IPv6 untuk mengakses website yang mengadopsi internet Protokol versi 4.

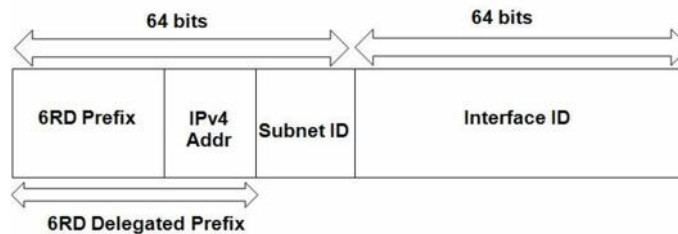


Gambar 2.5 6in4 Tunnel Broker

(Sumber: Shalini 2009)

2.5.2. 4in6 Tunnel Broker

4in6 Tunnel Broker merupakan kebalikan dari *6in4 Tunnel Broker* yang mengenkapsulasi paket IPv4 kedalam IPv6.



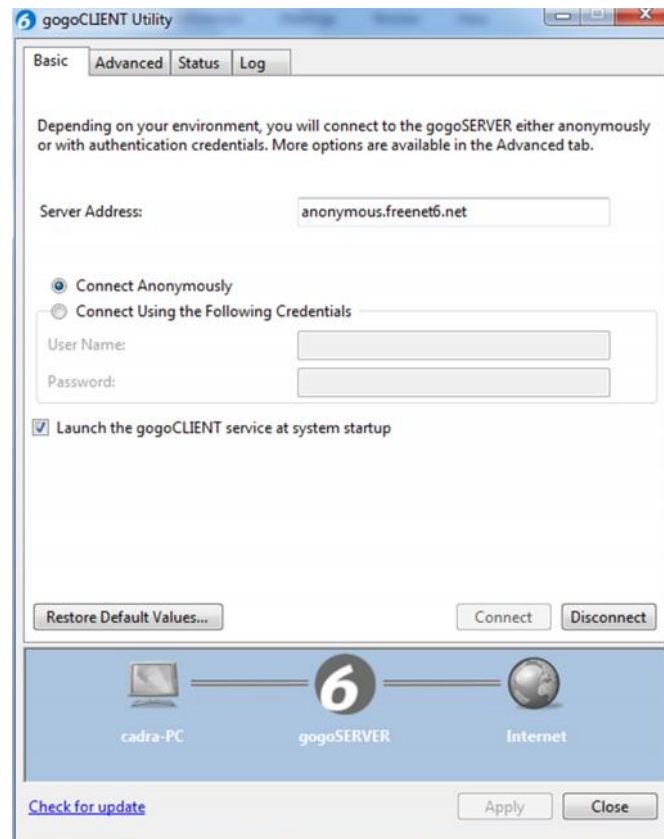
Gambar 2.6 4in6 Tunnel broker

2.6. Gogo6 Tunnel Broker

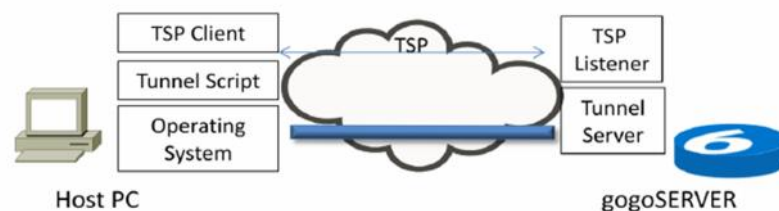
Gogo6 atau Freenet merupakan salah satu *tunnel broker* terbesar dari berbagai macam *tunnel broker* yang ada. *Tunnel broker* ini memiliki karakteristik seperti: menggunakan *software Tunnel Setting Protocol (TSP)*, *6in4* dan *4in6 Tunnel* juga konfigurasi TSP otomatis. Namun disisi lain gogo6 juga memiliki kelemahan seperti konfigurasi TSP tidak dapat dilakukan secara *manual* sehingga tidak bisa diaplikasikan pada *wireless router* yang mendukung tunneling IPv6.

Syarat dalam pengimplementasiannya, user harus memiliki akun gogo6 terlebih dahulu, setelah itu men-download *software gogoCLIENT* sebagai TSP pada *host*. Gambar 2.7 menunjukkan tampilan *gogoCLIENT* versi 1.2. Dari berbagai *tunnel broker* yang disediakan di internet, gogo6 ini merupakan *tunnel broker* yang konsisten dan masih aktif hingga saat ini dibanding kebanyakan *tunnel broker* konvensional yang tidak lagi bekerja dengan baik.

Gambar 2.8 mendeskripsikan sebuah skema dari *tunnel broker* yang bernama gogo6 ini dimana TSP *client* beroperasi diatas *operating system* yang mana pada penelitian menggunakan Microsoft Windows XP SP3. Antara client dan *server* harus saling terhubung untuk dapat mengakses IPv6 diatas infrastruktur IPv4. Adapaun gogoCLIENT ini dapat berjalan pada jaringan yang beroperasi dibelakang *Network Address Translation* (NAT) yang dapat dilihat pada gambar 2.9.

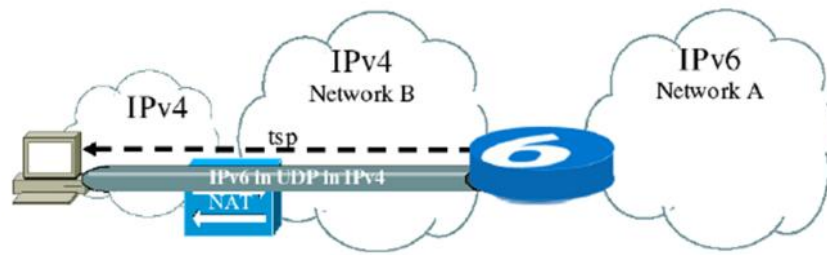


Gambar 2.7. GUI gogoCLIENT



Gambar 2.8. gogoCLIENT Component

(Sumber: gogoCLIENT Guide)

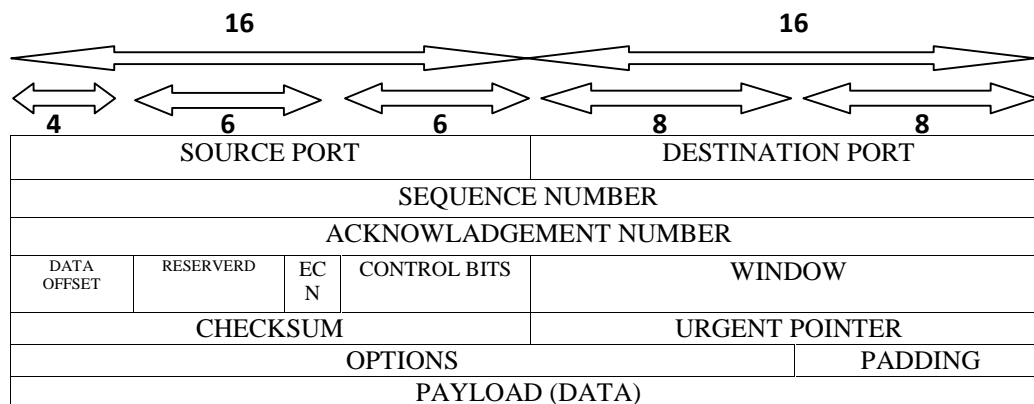


Gambar 2.9. gogoCLIENT dibelakang NAT

(Sumber: gogoCLIENT Guide)

2.7. Transfer Control Protocol (TCP)

Transmission Control Protocol (TCP) adalah salah satu jenis protokol yang memungkinkan kumpulan komputer untuk berkomunikasi dan bertukar data didalam suatu jaringan. TCP merupakan suatu protokol yang berada di lapisan transpor baik itu dalam tujuh lapis model referensi OSI atau model DARPA yang berorientasi sambungan (*connection-oriented*) dan dapat diandalkan (*reliable*). TCP dipakai untuk aplikasi-aplikasi yang membutuhkan keandalan data. (Sumber: Bahrul Ulum, 2012)



Gambar 2.10. Header TCP

(Sumber: <http://www.skullbox.net/tcpudp.php>)

Karakteristik dari TCP antara lain yaitu :

1. *Reliable* berarti data ditransfer ke tujuannya dalam suatu urutan seperti ketika dikirim.
2. Berorientasi sambungan (*connection-oriented*): Sebelum data dapat ditransmisikan antara dua *host*, dua proses yang berjalan pada lapisan aplikasi harus melakukan negosiasi untuk membuat sesi koneksi terlebih dahulu. Koneksi TCP ditutup dengan menggunakan proses terminasi koneksi TCP (*TCP connection termination*).

3. *Full-duplex*: Untuk setiap *host* TCP, koneksi yang terjadi antara dua *host* terdiri atas dua buah jalur, yakni jalur keluar dan jalur masuk. Dengan menggunakan teknologi lapisan yang lebih rendah yang mendukung *full-duplex*, maka data pun dapat secara simultan diterima dan dikirim. *Header* TCP berisi nomor urut (*TCP sequence number*) dari data yang ditransmisikan dan sebuah *acknowledgment* dari data yang masuk
4. Memiliki layanan *flow control*: Untuk mencegah data terlalu banyak dikirimkan pada satu waktu, yang akhirnya membuat *macet* jaringan *internetwork* IP, TCP mengimplementasikan layanan *flow control* yang dimiliki oleh pihak pengirim yang secara terus menerus memantau dan membatasi jumlah data yang dikirimkan pada satu waktu. Untuk mencegah pihak penerima untuk memperoleh data yang tidak dapat disangganya (*buffer*), TCP juga mengimplementasikan *flow control* dalam pihak penerima, yang mengindikasikan jumlah *buffer* yang masih tersedia dalam pihak penerima.
5. Melakukan segmentasi terhadap data yang datang dari lapisan aplikasi dalam DARPA Reference Model.
6. Mengirimkan paket secara *one-to-one*: hal ini karena memang TCP harus membuat sebuah sirkuit logis antara dua buah protokol lapisan aplikasi agar saling dapat berkomunikasi. TCP tidak menyediakan layanan pengiriman data secara *one-to-many*.

Adapun langkah-langkah cara kerja dari protokol TCP/IP ini adalah :

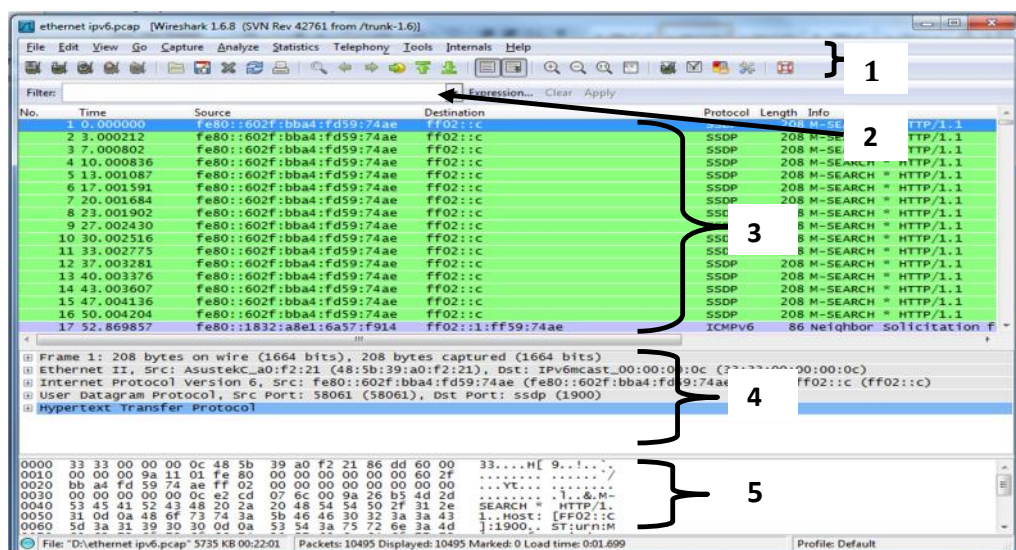
1. Pertama, datagram dibagi-bagi ke dalam bagian-bagian kecil yang sesuai dengan ukuran *bandwith* dimana data tersebut akan dikirimkan.
2. Pada lapisan TCP, data tersebut lalu dibingkai dengan informasi *header* yang dibutuhkan. Misalnya seperti cara mengarahkan data tersebut ke tujuannya, cara merangkai kembali kebagian-bagian data tersebut jika sudah sampai pada tujuannya, dan sebagainya.
3. Setelah *datagram* dibungkus dengan *header* TCP, datagram tersebut dikirim kepada lapisan IP.
4. IP menerima *datagram* dari TCP dan menambahkan *headernya* sendiri pada *datagram* tersebut.
5. IP lalu mengarahkan *datagram* tersebut ke tujuannya.

6. Komputer penerima melakukan proses-proses perhitungan, ia memeriksa perhitungan *checksum* yang sama dengan data yang diterima.
7. Jika kedua perhitungan tersebut tidak cocok berarti ada *error* sewaktu pengiriman dan *datagram* akan dikirimkan kembali. (Sumber: <http://thelolbee.wordpress.com/2010/05/20/tcp-dan-udp>, 2010)

2.8. Monitoring dengan Wireshark

Wireshark adalah salah satu dari sekian banyak *tool Network Analyzer* yang banyak digunakan oleh *Network administrator* untuk menganalisa kinerja jaringannya. Wireshark banyak dipilih karena *interface*-nya yang menggunakan GUI atau tampilan grafis. Awalnya bernama Ethereal, pada mei 2006 proyek ini berganti nama menjadi Wireshark karena masalah merek dagang. Wireshark mampu menangkap paket-paket data/informasi dalam jaringan. Semua jenis paket informasi dalam berbagai format protokol pun akan dengan mudah ditangkap dan dianalisis. Selain sebagai *analyzer*, wireshark juga sering digunakan sebagai alat *snipping* yang memungkinkan *network administrator* memantau apa yang dilakukan *user* lain pada sebuah jaringan yang sama. Struktur dari *wireshark graphical user interface* yang terlihat pada Gambar 2.11 adalah sebagai berikut :

1. *Command menu*
2. *Display filter specification* : untuk memfilter paket data
3. *Listing of captured packets* : paket data yang tertangkap oleh wireshark
4. *Details of selected packet header* : data lengkap tentang *header* dari suatu paket
5. *Packet contents* : isi dari suatu paket data



Gambar 2.11. Layar utaman Wireshark

BAB III

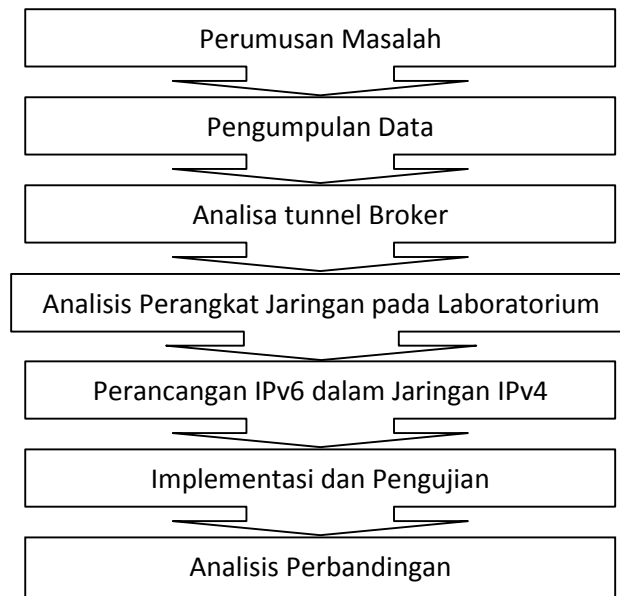
METODE PENELITIAN

3.1. Jenis Penelitian

Metodologi yang digunakan dalam penelitian yang berjudul “Analisis dan Implementasi interkoneksi IPv6 dan IPv4 Pada Aplikasi *Tunnel Broker* di Laboratorium Jaringan Komputer” adalah penelitian studi kasus dimana dilakukan penerapan sebuah jaringan IPv6 yang terenkapsulasi di dalam paket-paket data IPv4 dengan menggunakan mekanisme transisi *tunneling* yang pengaplikasiannya menggunakan *tunnel broker*.

Selain implementasi, pada penelitian ini juga dilakukan analisis kinerja dari proses *tunneling* tersebut. Waktu yang terpakai dalam pengiriman data pada *tunnel broker* akan dibandingkan dengan waktu yang diperlukan saat pengiriman data menggunakan IPv4.

3.2. Tahapan Penelitian



Gambar 3.1 Diagram blok desain penelitian

3.2.1. Perumusan Masalah

Merumuskan masalah tentang bagaimana mengimplementasikan *suatu tunnel broker* IPv6 pada di laboratorium komputer dan analisis kualitas waktu saat mentransfer data yang dikirim maupun diterima.

3.2.2. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data tentang implementasi dan analisis *tunnel broker* di laboratorium jaringan komputer. Semua tahap pada proses pengumpulan data tersebut diperoleh dari :

a. Studi Pustaka

Studi pustaka dilakukan dengan cara mempelajari buku-buku, jurnal, artikel dan forum-forum yang berhubungan dengan permasalahan yang dibahas yaitu tentang bagaimana menjalankan pengalamatan IPv6 yang terenkapsulasi IPv4 pada sebuah lingkup jaringan.

b. Observasi

Observasi atau pengamatan langsung merupakan metode pengumpulan data dengan menggunakan indera penglihatan untuk melihat dan mengidentifikasi permasalahan interkoneksi IPv6 dengan IPv4 di Laboratorium Jaringan Komputer.

3.2.3. Analisis *Tunnel Broker*

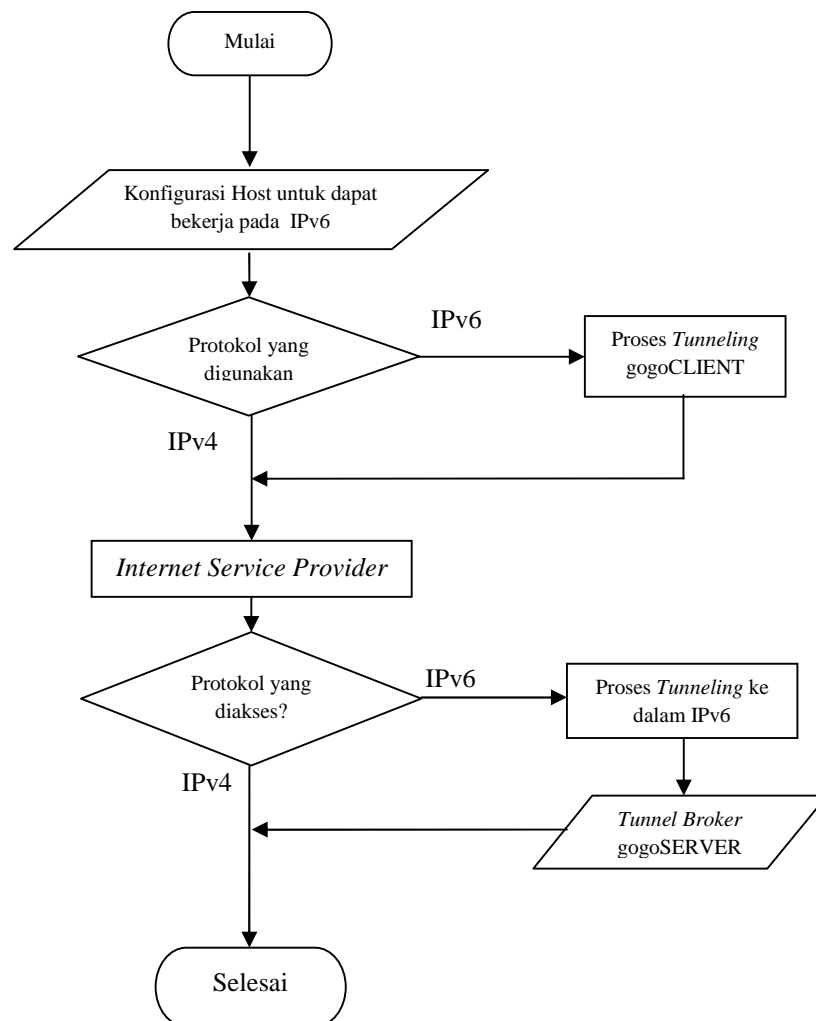
Analisa *tunnel broker* dilakukan untuk menyeleksi *tunnel broker* yang tepat untuk implementasi. Alasan menggunakan Indonesia IPv6 sebagai proses *tunneling* adalah sebagai berikut:

1. Proses konfigurasi pada *client* dengan *software* GUI
2. Posisi *tunnel server* yang lebih dekat dengan *client* mengurangi waktu transmisi sehingga waktu yang dibutuhkan untuk mengirim dan menerima data menjadi relatif singkat

3.2.4. Perancangan IPv6 dalam Jaringan IPv4

Dalam membuat sebuah implementasi perlu adanya perancangan maupun bayangan kerja yang akan dibuat. Dan rancangan penerapan IPv6 di dalam Laboraturium Jaringan Komputer adalah sebagai berikut.

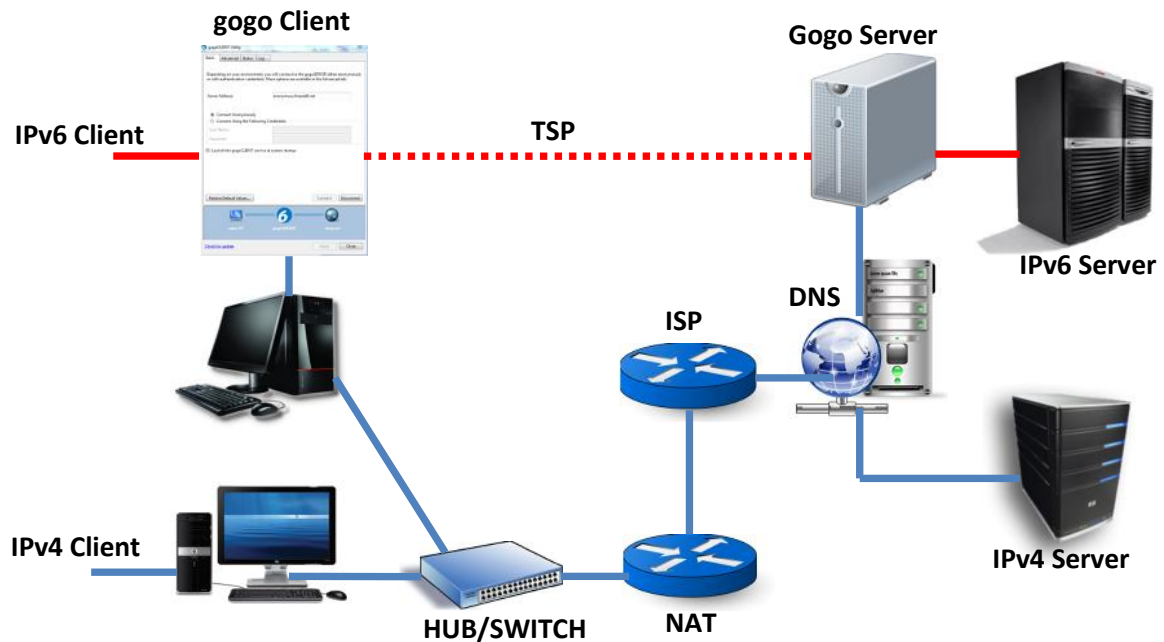
3.2.4.1. Diagram Alir (*Flow Chart*) Perancangan Interkoneksi IPv6 dan IPv4



Gambar 3.2 *Flow Chart* implementasi interkoneksi IPv4 ke IPv6

Flow Chart diatas menggambarkan paket-paket IPv6 dienkapsulasi ke dalam IPv4 agar dapat melewati ISP yang berbasis IPv4. Untuk jaringan yang menggunakan infrstruktur IPv4 maka paket datanya tidak dienkapsulasi karena yang menjadi protokol pembawanya adalah IPv4 itu sendiri.

3.2.4.2. Skema Perancangan Interkoneksi IPv6 dan IPv4 dengan Aplikasi *Tunnel Broker*



Gambar 3.3 Skema Implementasi IPv6 di dalam Jaringan IPv4

Host yang memiliki pengalamatan IPv4 tidak perlu melewati *tunneling* karena basis pada skema ini adalah IPv4, namun untuk IPv6 harus menggunakan *tunneling* karena IPv6 tidak bisa begitu saja melewati jaringan IPv4 tanpa proses transisi. Kemudian ISP yang tersambung langsung ke Internet memiliki peranan sebagai perantara komunikasi jaringan WAN. Selanjutnya informasi data akan disampaikan ke *server Tunnel Broker* untuk dapat mengenkapsulasi data IPv4 ke IPv6.

3.2.5. Analisis Perangkat Jaringan pada Laboratorium

Tahap ini adalah survei peralatan yang akan digunakan untuk penelitian. Adapun analisa perangkat jaringan yang digunakan dalam implementasi meliputi :

1. ISP yang akan diimplementasikan
2. Spesifikasi *host*
3. Jumlah *host* yang akan diuji
4. Sistem operasi pada *host*

3.2.6. Implementasi dan Pengujian

Implementasi adalah tahap dimana perancangan-perancangan yang dilakukan sebelumnya diterapkan secara nyata. Adapun langkah-langkahnya sebagai berikut:

1. Konfigurasi host
2. Registerasi ke situs *tunnel broker*
3. Instalasi dan aktivasi *tunnel broker*
4. Pengujian

Pengujian adalah tahapan dimana implementasi yang dijalankan diuji kelayakannya dan benar-benar berjalan dengan baik. Dalam pengujian ada beberapa cara yang digunakan untuk membuktikan bahwa implementasi yang dilakukan telah berhasil seperti Melakukan *ping* dan *browsing* untuk IPv4 dan IPv6.

3.2.7. Analisis Perbandingan

Pada tahap analisis diperlukan sebuah *software analyzer* untuk memonitor paket-paket data yang dikirim dari sebuah komputer menuju ke internet yang bernama Wireshark. Wireshark akan menangkap paket-paket data yang mengalir pada jalur transmisi secara detail. Data statistik yang didapat dari hasil *capturing* akan disaring. Aktifitas yang dianalisis adalah *download* dan *upload*.

Persentase paket transfer

$$\text{Persentase Paket Transfer} = \frac{\text{jumlah Paket Download}}{\text{Jumlah Paket yang Tersaring}} \times 100\% \quad \dots\dots\dots (3.1)$$

Delay Transfer

$$\text{Delay Transfer} = \text{Jumlah yang tersaring} \times \text{Persentase Paket Transfer} \quad \dots\dots\dots (3.2)$$

Penelitian ini membutuhkan 10 kali percobaan *download* dan *upload* melalui IPv4 dan 10 kali *download* dan *upload* melalui gogo6, hasil yang didapat harus dicari nilai rata-ratanya pada dengan rumus:

$$\text{Delay rata-rata} = \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n} \quad \dots\dots\dots (3.3)$$

Rumus selanjutnya adalah mencari perbandingan antara *delay* IPv4 dengan *delay*

gogo6 :

$$\text{Gogo6 : IPv4} = \frac{\text{gogo6}}{\text{IPv4}} : \frac{\text{IPv4}}{\text{IPv4}} \quad \dots\dots\dots (3.4)$$

BAB IV

IMPLEMENTASI DAN ANALISIS

4.1. Implementasi Sistem

Pada dasarnya sistem atau jaringan yang dibuat memerlukan prasarana penunjang agar implementasi yang diterapkan dapat beroperasi dengan baik. Ada dua kategori perangkat sebagai pendukung sistem dengan berbagai spesifikasinya antara lain:

Perangkat Keras:

1. PC Host
 - a. Tipe : *Desktop*
 - b. Prosesor : *Pentium 4 DualCore*
 - c. Ram : *512MB*
 - d. NIC : *Onboard*
 - e. WLAN : *Broadcom*



Gambar 4.1. *Personal Computer*

2. *Wireless Router*

Merek : TP-Link

Perangkat Lunak:

1. Sistem Operasi : Microsoft Windows XP SP3
2. *Tunnel Broker* : Gogo SERVER dan gogoCLIENT

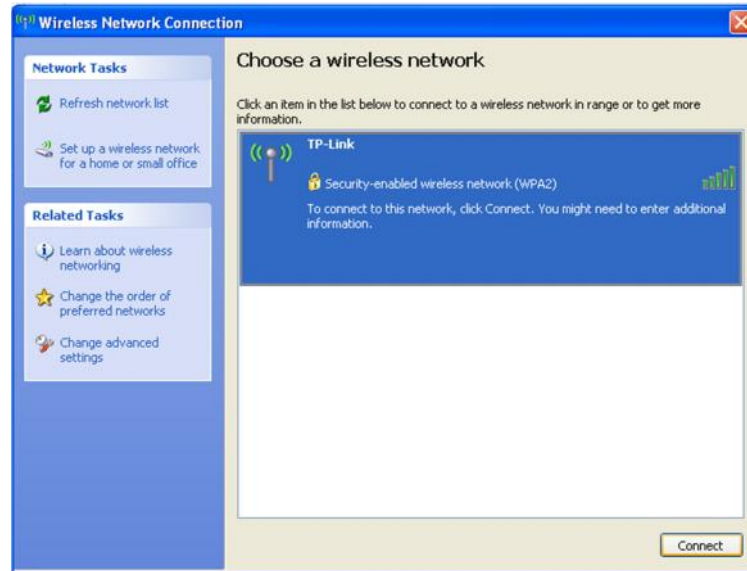
- 3. *Software Analisis* : Wireshark
- 4. *Software Pengujian* : *Command prompt*
: *Browser (Mozilla 20.0)*
- 5. ISP
 - a. Nama : Telkom Speedy
 - b. Teridentifikasi : NAT
 - c. IP Public : DHCP

Beberapa perangkat yang tertera sangat diperlukan untuk implementasi *tunnel broker*. Pada Laboratorium Komputer terdapat 19 buah PC dengan spesifikasi yang sama dimana *bandwidth* yang diterima sebesar 384Kbps, namun percobaan ini hanya digunakan dua PC dalam satu jaringan dengan sistem pengalamatan yang berbeda yaitu IPv4 tanpa menggunakan *tunnel broker* dan IPv6 menggunakan *tunnel broker gogo6*.

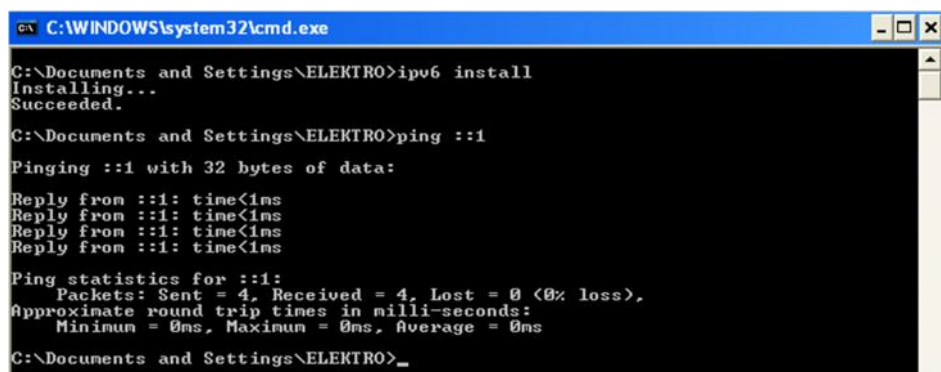
4.1.1. Konfigurasi Host

Host atau PC yang akan digunakan sebagai *end-point* dalam transisi jaringan ini harus sudah terkoneksi dengan internet terlebih dahulu. Maka diperlukan langkah-langkah konfigurasinya agar penelitian berjalan dengan baik. Berikut adalah langkah-langkahnya:

- a. Aktifkan kartu jaringan nirkabel yang berada di PC.
- b. Lakukan *Searching*, jika *hotspot* ditemukan hubungkan dengan menekan tombol *Connect* pada jendela *Wireless Network Connection* dapat dilihat pada gambar 4.2.
- c. Aktifkan fasilitas IPv6 dengan cara masuk ke *command prompt* kemudian ketikkan “*ipv6 install*”, setelah proses berhasil lakukan *ping* ke alamat *loopback* dengan cara ketik “*ping ::1*” jika respon didapat maka fasilitas IPv6 pada windows XP telah diaktifkan seperti yang dapat dilihat pada gambar 4.3.
- d. IPv6 sudah dapat diimplementasikan pada *Local Area Network*.



Gambar 4.2. *Wireless Network Connection*



Gambar 4.3. *Command Prompt*

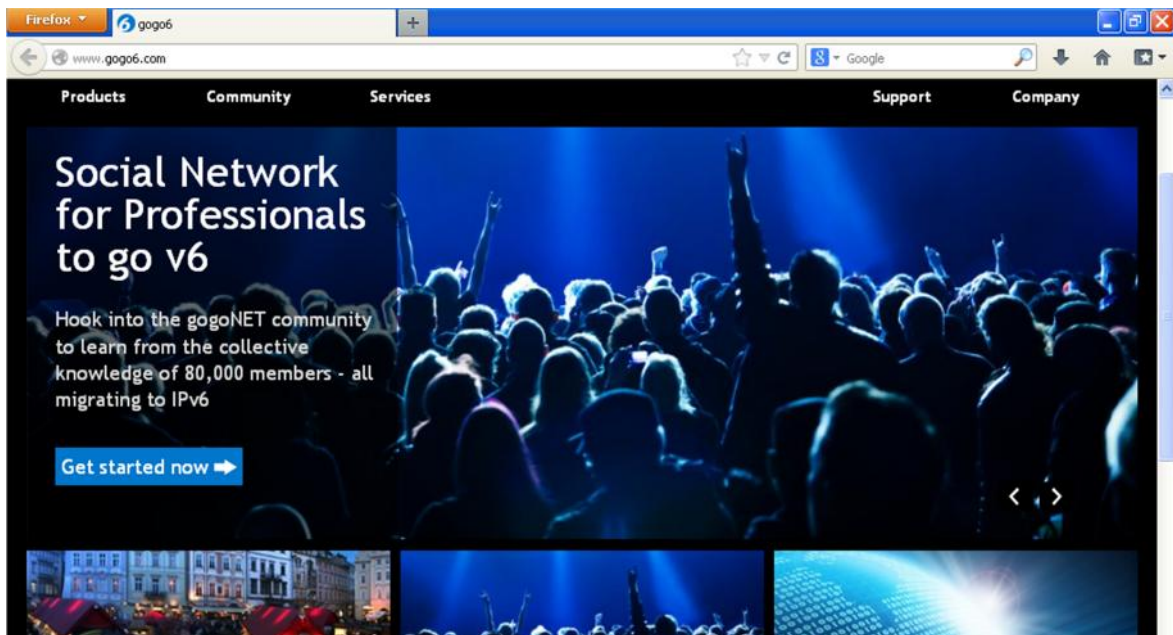
4.1.2. Konfigurasi *Tunnel Broker*

Sebelum menggunakan *tunnel broker* sebagai aplikasi interkoneksi antara IPv6 dengan IPv4 diperlukan registrasi akun *tunnel broker* agar mendapat alamat dari *server tunnel broker*. Langkah-langkah yang digunakan adalah sebagai berikut:

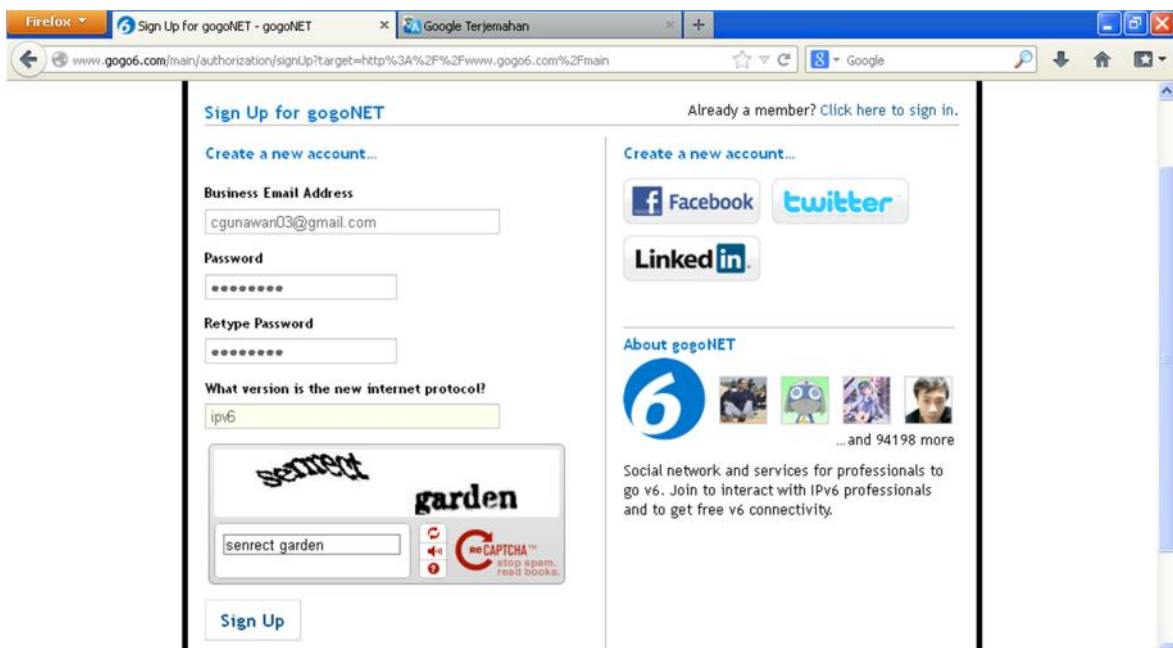
1. Kunjungi www.gogo6.com seperti gambar 4.4 lalu klik “sign up” untuk mendaftar atau registrasi untuk mendapatkan akun *tunnel broker* seperti gambar 4.5.
2. Klik *download*, pilih *gogoCLIENT 1.2 win32* karena sistem operasi yang digunakan adalah *Microsoft Windows XP* yang menggunakan 32bit seperti yang terlihat pada gambar 4.6. *Website* alternatif untuk untuk mendapatkan *software gogoCLIENT* ini juga bisa diakses di *provider tunnel broker* sejenisnya

<http://www.ipv6now.com.au/downloads.php>. alamat ini digunakan jika *software* yang di-download pada server gogo6 mengalami kerusakan atau *error*.

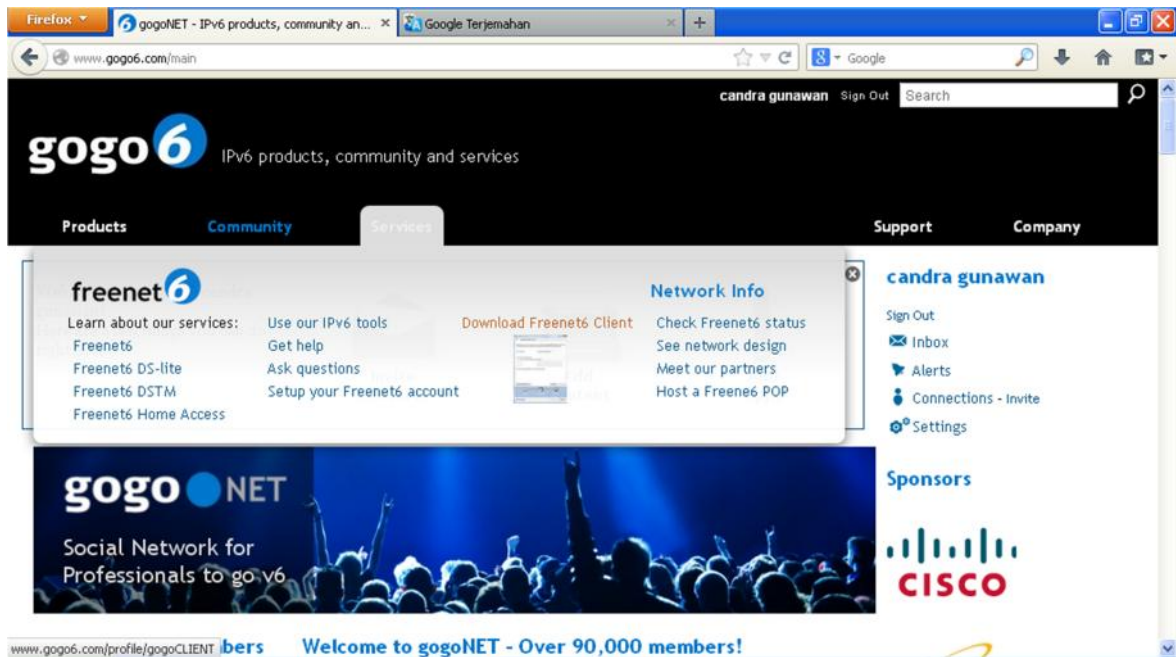
3. Instal *software tunnel broker* tersebut seperti gambar 4.8 dan gambar 4.9 kemudian jalankan *software* tersebut, jika keluar tampilan *Connected to gogoSERVER* maka *host* telah terhubung dengan gogoSERVER seperti yang terlihat pada gambar 4.10.
4. Jika *tunnel broker* berjalan dengan baik, pengujian dapat dilakukan.



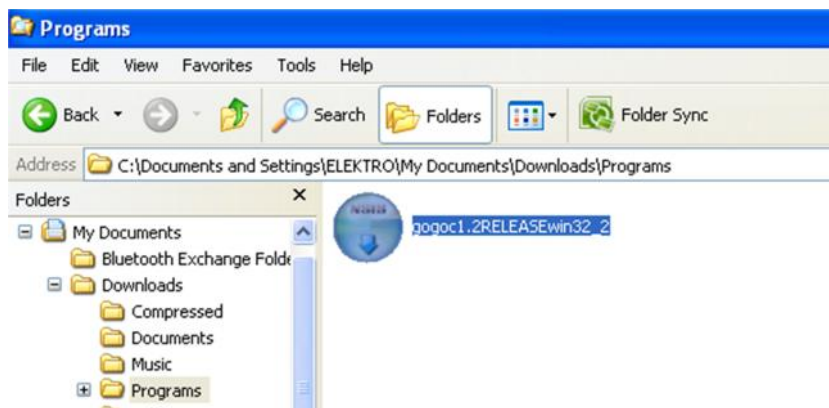
Gambar 4.4. Halaman *dashboard* gogo6



Gambar 4.5. Halaman *Sign Up* gogo6



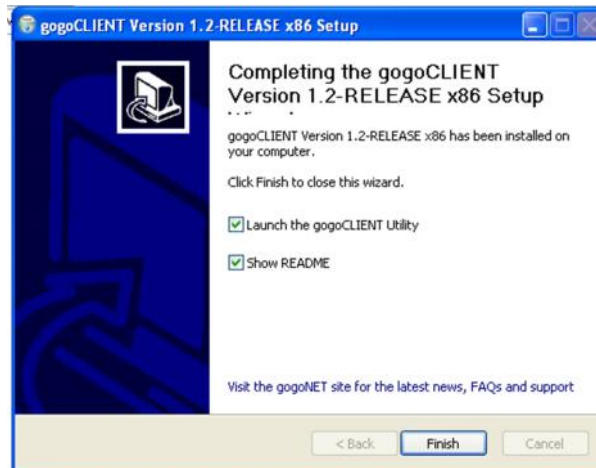
Gambar 4.6. Halaman *download* gogoCLIENT



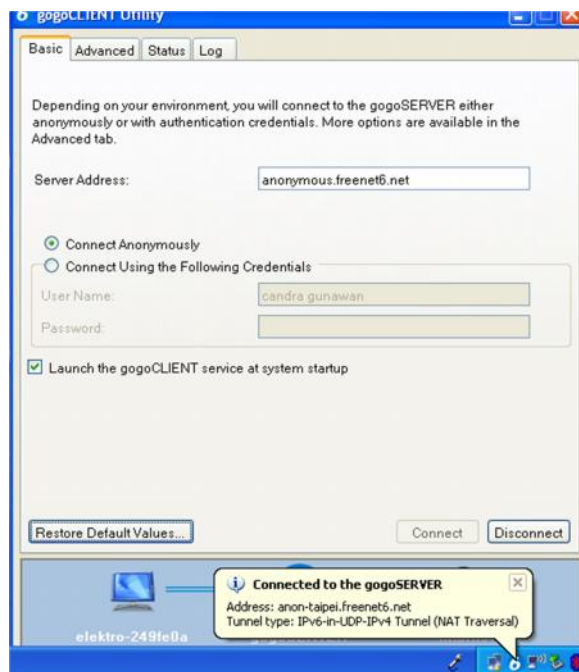
Gambar 4.7. *Software* gogoCLIENT yang telah didownload



Gambar 4.8. Mulai instalasi gogoCLIENT



Gambar 4.9. Instalasi gogoCLIENT selesai



Gambar 4.10. Interface gogoCLIENT

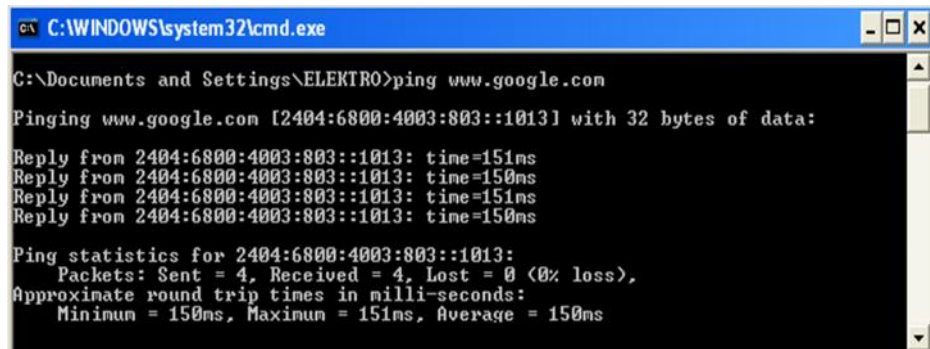
4.1.3. Pengujian Sistem

Pengujian dilakukan untuk mengindikasikan apakah implementasi berjalan lancar sesuai dengan yang diharapkan atau masih terjadi kesalahan-kesalahan yang tidak diinginkan. Beberapa macam pengujian sistem dapat dijabarkan sebagai berikut:

4.1.3.1 Pengujian dengan *Ping*

Ping atau memberikan umpan kepada *server* biasa dilakukan *user* untuk mengetahui apakah sebuah jaringan yang dibuat atau yang dijalankan saat itu berjalan dengan baik atau tidak. Dengan dengan melakukan *ping*, dapat dilihat lama waktu transfer paket ICMP ke

server maupun *host* lain dalam jaringan itu sendiri sehingga didapat hasil berupa waktu maksimal, waktu minimal, waktu rata-rata dan total paket yang dikirim juga jenis protokol yang digunakan, dalam hal ini IP Address yang diidentifikasi adalah IPv6.



Gambar 4.11. Pengujian dengan *ping*

4.1.3.2. Pengujian dengan *browser*

Browser adalah cara kedua untuk pengujian. Pada pengujian ini peneliti mencoba mengakses *website* yang memiliki pengalamatan IPv6 murni bukan merupakan *dualstack*. Salah satu *website* yang murni menggunakan IPv6 antara lain Google (ipv6.google.com) pada gambar 4.12 dan Facebook (www.v6.facebook.com) pada gambar 4.13, alasannya tidak semua *tunnel broker* yang dapat menampilkan proses *ping* dapat juga mengakses *browser* dengan baik, selain itu jika aplikasi *tunnel broker* belum bekerja sempurna atau dengan kata lain IPv6 masih belum aktif maka *website* tersebut tidak dapat ditampilkan karena kedua *website* ini hanya memiliki satu alamat internet protokol saja yaitu IPv6.



Gambar 4.12. ipv6.google.com



Gambar 4.13. www.v6.facebook.com

4.2. Analisis Perbandingan *Delay*

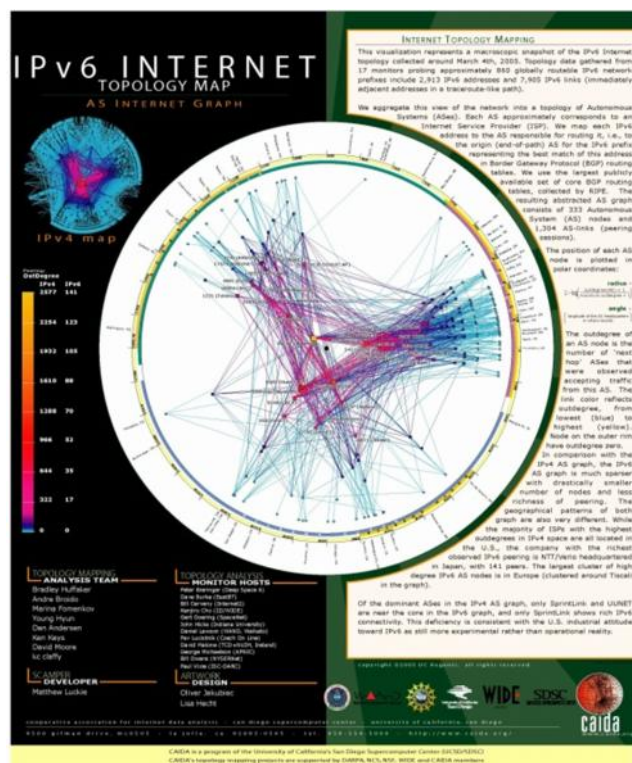
Delay merupakan waktu yang dibutuhkan dalam proses pengiriman maupun penerimaan data demi data dalam suatu jaringan internet. Paket-paket data yang dikirim menggunakan pengalamatan IPv4 tentu berbeda dengan pengalamatan IPv6, selain *range* alamat yang berbeda juga pada IPv6 ini dienkapsulasi dengan aplikasi *tunnel broker*. Untuk membandingkan keduanya maka penulis melakukan analisis perbandingan *delay* antara IPv4 dengan *tunnel broker* gogo6 dengan parameter yang dianalisis seperti analisis *Download* dan analisis *Upload*.

4.2.1. Analisis *Delay* Aktifitas *Download*

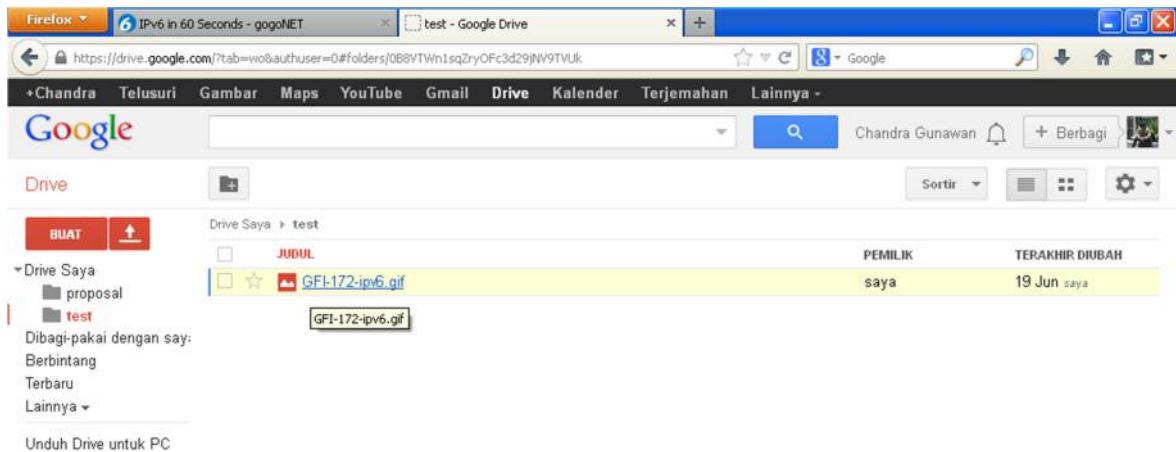
Aktifitas *download* memiliki perbedaan *delay* antara IPv4 dengan IPv6. Untuk melakukan analisis *download* diperlukan langkah-langkah sebagai berikut:

1. Buka *browser* mozilla dan ketik www.google.com kemudian pilih *drive*. Peneliti menggunakan akun google pribadi untuk analisis ini.
2. Pada percobaan ini penulis men-*download* file berektensi JPG dengan ukuran *file* sebesar 1.244.473 *byte*, dapat dilihat pada gambar 4.14
3. Buka wireshark kemudian *capture* pada *interface* IPv4.

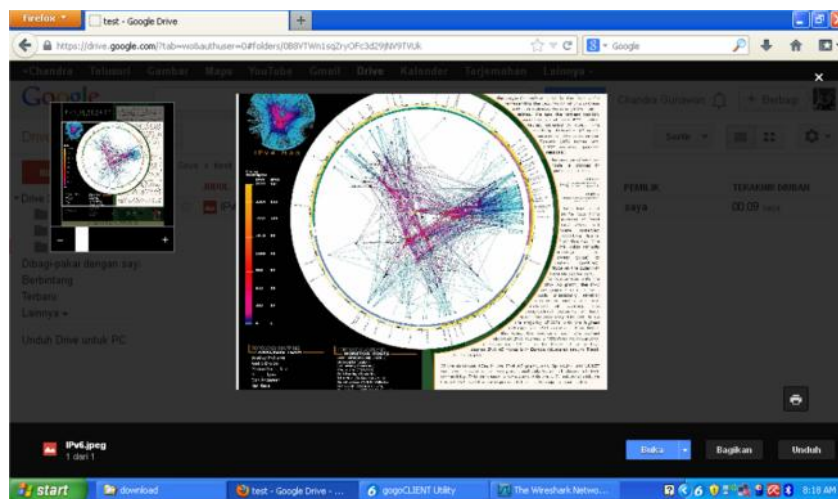
4. Klik tombol unduh pada *browser* yang sedang dalam proses *download* yang dapat dilihat pada gambar 4.16, jika selesai maka *capturing* pada wireshark dapat dihentikan dan hasil dapat dianalisis seperti yang terlihat pada gambar 4.21.
5. Untuk mencari *delay download*, yang harus dilakukan adalah pada *tab* menu pilih *statistics>Conversations* maka akan terlihat data statistik pada tiap paket protokol. Pilih protokol TCP seperti yang terlihat pada gambar 4.23, maka jumlah paket dan kapasitas paket yang paling besar merupakan *file* yang telah ditransfer.
6. Untuk memastikan bahwa data yang dituju merupakan *file* yang ditransfer, klik data tersebut kemudian klik *Follow Stream* maka akan keluar jendela *Follow TCP Stream* yang menunjukkan kapasitas data sebesar 1.271.494 *Byte* seperti yang terlihat pada gambar 4.24 . Nilai ini mendekati besar ukuran file yang ditransfer yaitu 1.244.473 *Byte*. kapasitas yang ditunjukkan wireshark belum dinormalisasi sehingga hasil yang ditampilkan tidak sama dengan kapasitas *file* semula.
7. Lakukan kembali langkah 1 sampai 6 dengan menggunakan *interface* gogo6.



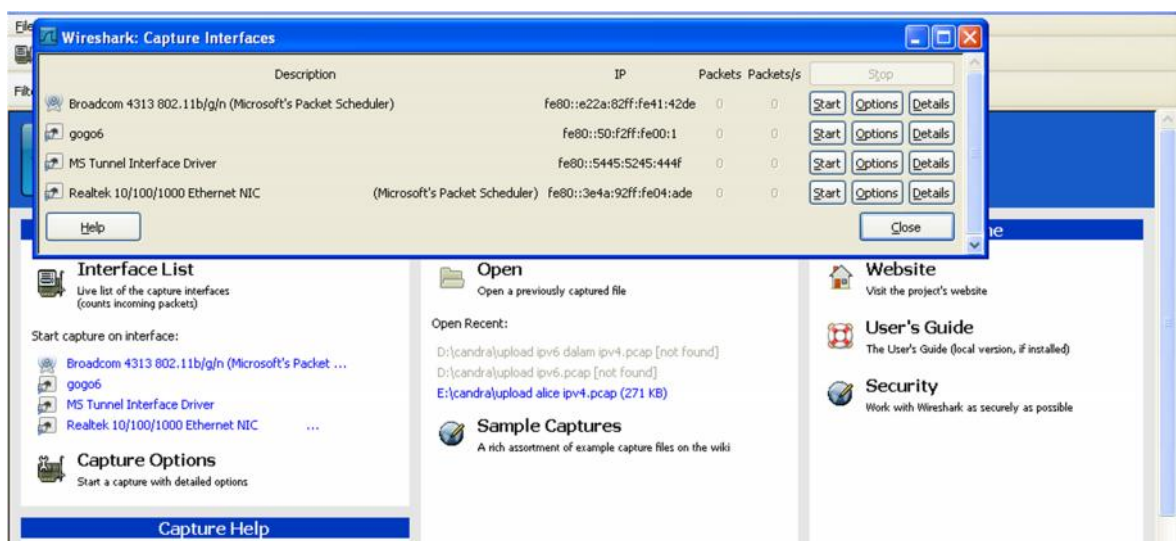
Gambar 4.14. File JPG yang akan ditransfer



Gambar 4.15. Google drive sebagai *interface download*



Gambar 4.16. View sebelum download



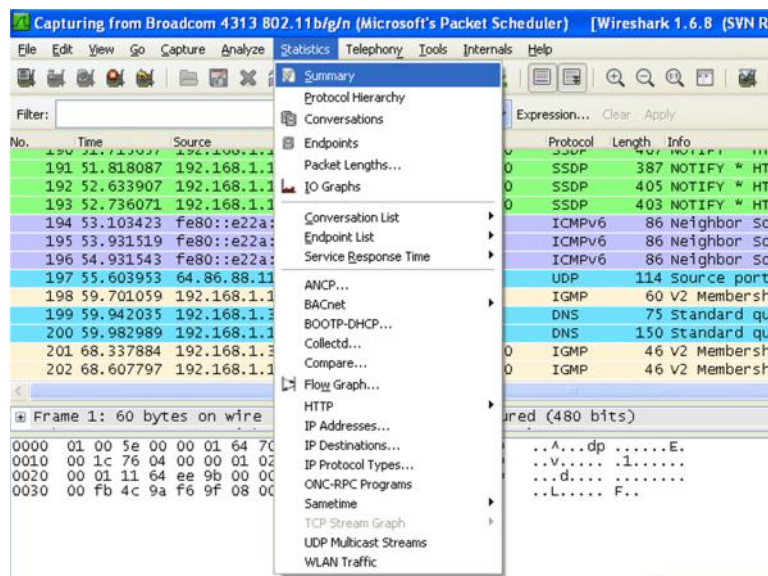
Gambar 4. 17. Pemilihan *interface* untuk proses *capture*



Gambar 4.18. Proses *Download* sedang berjalan



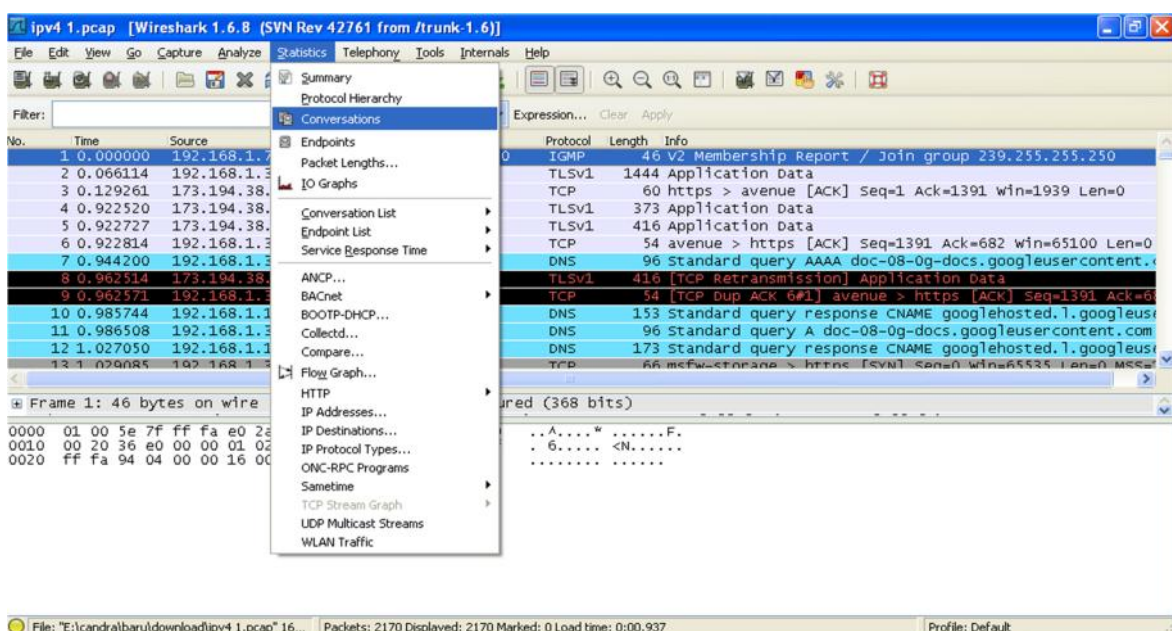
Gambar 4.19. File selesai di-download



Gambar 4.20. Menu *Statistics* pada wireshark



Gambar 4.21. Hasil *capturing download* pada wireshark



Gambar 4.22. Menu menuju *Conversations*

$Delay\ download = \text{Jumlah yang tersaring} \times \text{Persentase Paket Download}$

$Delay\ Download = 40,3567\text{second} \times 56,21\%$

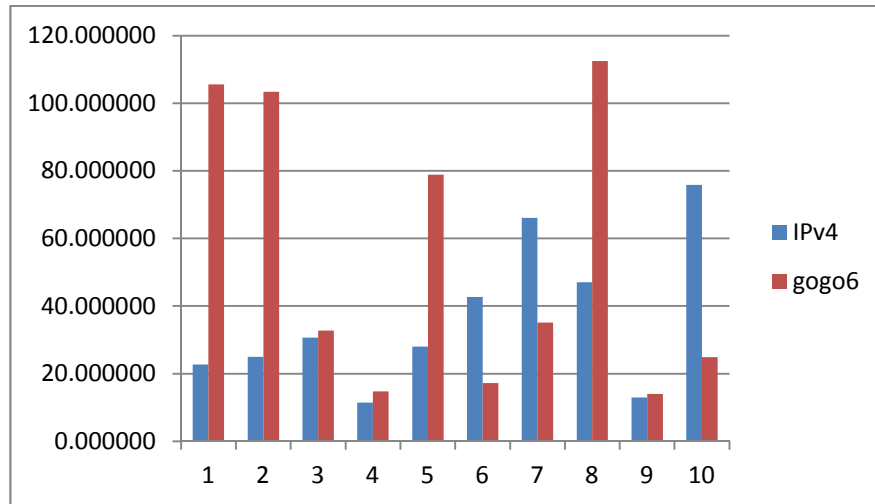
= 22,685576 second

Jadi, delay murni pada sebuah percobaan *download* untuk percobaan pertama ialah 22,685576 *second*. Nilai *delay* tidak *absolute* karena dipengaruhi banyak faktor dan kondisi jaringan pada saat pengiriman paket data.

Berdasarkan rumus di atas, penulis melakukan percobaan sebanyak 10 kali untuk mendapatkan *delay* rata-rata yang akurat karena jika hanya satu kali percobaan saja belum bisa membuktikan kualitas gogo6 dengan IPv4 secara keseluruhan hasil percobaan dapat dilihat pada tabel 4.1.

Tabel 4.1. Data *download* dengan wireshark

Interface	No. Percobaan	Jumlah Paket Total (Packet)	Jumlah Paket yang Tersaring (Packet)	Delay yang Disaring (Second)	Jumlah Paket Recieve (packet)	Persentase Paket Recieve (%)	Delay Download Murni (Second)
IPv4	1	2170	1674	40,3567	941	56,21	22,685576
	2	2054	1643	44,1412	931	56,66	25,012451
	3	2243	1707	55,1669	950	55,65	30,702141
	4	1704	1659	20,5657	926	55,82	11,479107
	5	2105	1679	50,2696	937	55,81	28,053970
	6	2259	1633	75,0056	930	56,95	42,715988
	7	2276	1678	120,4688	920	54,83	66,049640
	8	1891	1688	84,6214	939	55,63	47,073160
	9	1689	1617	22,5746	929	57,45	12,969575
	10	2035	1720	137,8477	946	55,00	75,816235
gogo6	1	2625	1818	175,6876	1092	60,07	105,528525
	2	2179	1894	176,6561	1108	58,50	103,344751
	3	1720	1684	51,8077	1065	63,24	32,764371
	4	1673	1655	23,1160	1057	63,87	14,763512
	5	1955	1871	134,3432	1099	58,74	78,911372
	6	1692	1669	26,8650	1068	63,99	17,191025
	7	1747	1707	56,0177	1070	62,68	35,113614
	8	2083	1932	196,6666	1105	57,19	112,482709
	9	1672	1651	21,9486	1052	63,72	13,985419
	10	1694	1665	39,0933	1060	63,66	24,888227



Gambar 4.25. Grafik *delay* tiap percobaan *download*

Dilakukan 10 percobaan dengan cara yang sama untuk menemukan hasil yang akurat pada penelitian ini. Balok diagram berwarna biru menggambarkan *delay* yang didapat oleh IPv4 , sedangkan balok merah menggambarkan *delay* pada gogo6. Dari hasil tersebut maka ditentukan *delay* murni rata-rata untuk mewakili pada 10 kali percobaan proses *download*.

$$\begin{aligned}
 \text{Delay download rata-rata IPv4} &= \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n} \\
 \text{Delay download rata-rata IPv4} &= \frac{22,685576+ 25,012451+ 30,702141+ 11,479107+ 28,053970+ 42,715988+ 66,049640+ 47,073160+ 12,969575+ 75,816235}{10} \\
 &= \mathbf{36,255784 \text{ seconds}} \\
 \text{Delay download rata-rata gogo6} &= \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n} \\
 \text{Delay download rata-rata gogo6} &= \frac{105,528525+ 103,344751+ 32,764371+ 14,763512+ 78,911372+ 17,191025+ 35,113614+ 112,482709+ 13,985419+ 24,888227}{10} \\
 &= \mathbf{53,897352 \text{ seconds}}
 \end{aligned}$$

Jadi, *delay* rata-rata pada proses *download* keseluruhan pada IPv4 adalah 36,255784 *seconds* dan *delay* rata-rata pada proses *download* keseluruhan *tunnel broker* gogo6 adalah 53,897352 *seconds*.

Jika diamati lebih jauh, ada perbedaan signifikan dari jumlah paket *recieve* antara percobaan *download* IPv4 dengan gogo6 pada tabel 4.1. untuk menentukan rata-rata jumlah paket *recieve* maka digunakan rumus sebagai berikut :

$$\text{Jumlah paket recieve rata-rata IPv4} = \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n}$$

$$\begin{aligned}\text{Jumlah paket receive rata-rata IPv4} &= \frac{941+931+950+926+937+930+920+939+929+946}{10} \\ &= \mathbf{934,9\ packets}\end{aligned}$$

$$\text{Jumlah paket receive rata-rata gogo6} = \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n}$$

$$\begin{aligned}\text{Jumlah paket receive rata-rata gogo6} &= \frac{1092+1108+1065+1057+1099+1068+1070+1105+1052+1060}{10} \\ &= \mathbf{1077,6\ packets}\end{aligned}$$

Jadi, jumlah paket *receive* rata-rata pada proses *download* keseluruhan pada IPv4 adalah 934,9 *packets* dan jumlah paket *receive* rata-rata pada proses *download* keseluruhan *tunnel broker* gogo6 adalah 1077,6 *packets*.

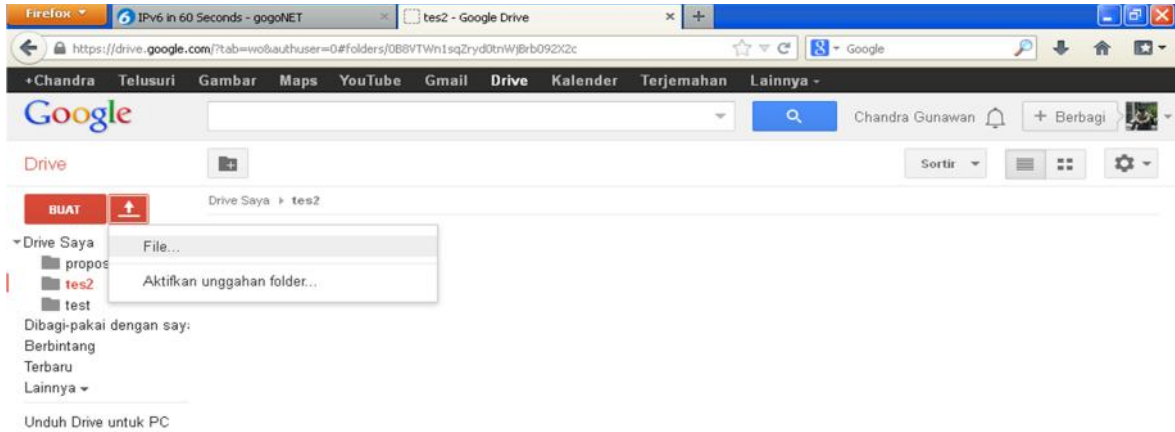
4.2.2. Analisis Delay Aktivitas Upload

Aktivitas *upload* juga membutuhkan waktu dalam implementasinya, jika aktivitas *download* memiliki perbedaan *delay* antara IPv4 dengan IPv6 maka begitu pula dengan *upload*. Untuk melakukan analisis *upload* diperlukan langkah-langkah sebagai berikut:

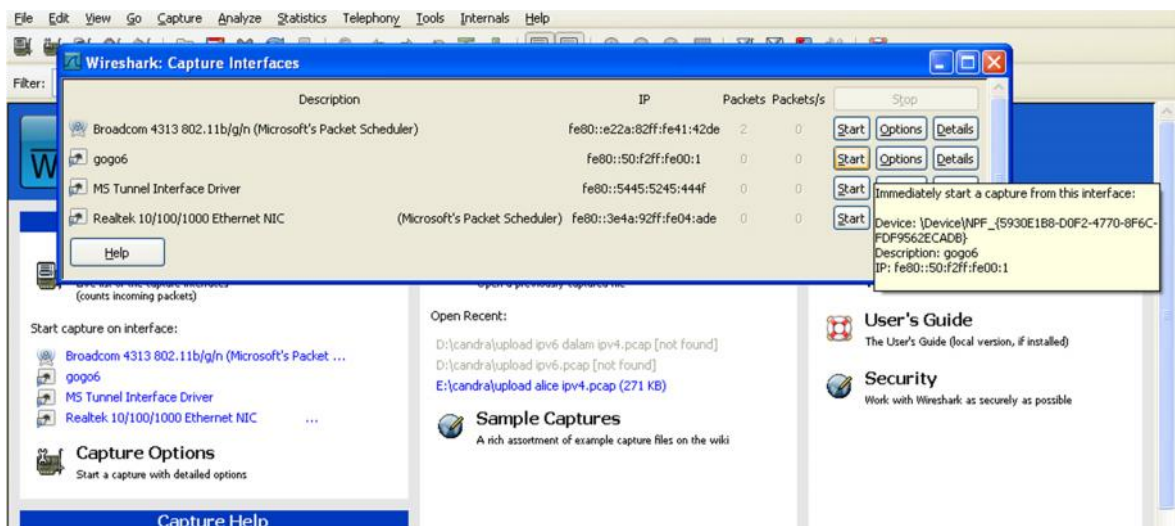
1. Buka *browser* mozilla dan ketik www.google.com kemudian pilih *drive*. Peneliti menggunakan akun google pribadi untuk analisis ini.
2. Pada percobaan ini penulis meng-*upload* file yang sama dengan *download* yaitu *file* ekstensi JPG dengan ukuran *file* sebesar 1.244.473 byte, dapat dilihat pada gambar 4.23
3. Buka wireshark kemudian *capture* pada *interface* IPv4 seperti gambar 4.27.
4. Klik tombol yang berwarna merah dan bergambar panah ke atas pada *browser* yang dapat dilihat pada gambar 4.28, kemudian pilih *file* yang akan di-*upload* seperti gambar 4. Jika selesai maka *capturing* pada wireshark dapat dihentikan dan hasil dapat dianalisis seperti yang terlihat pada gambar 4.29.
5. Untuk mencari *delay upload*, yang harus dilakukan adalah pada *tab* menu pilih *statistics>Conversations* maka akan terlihat data statistik pada tiap paket protokol. Pilih protokol TCP seperti gambar 4.32, maka jumlah paket dan kapasitas paket yang paling besar merupakan *file* yang telah ditransfer.
6. Untuk memastikan bahwa data yang dituju merupakan *file* yang ditransfer, klik data tersebut kemudian klik *Follow Stream* maka akan keluar jendela *Follow TCP Stream* yang menunjukkan kapasitas data sebesar 1.256.827 *Byte* seperti yang terlihat pada gambar 4.33. Nilai ini mendekati besar ukuran *file* yang ditransfer

yaitu 1.244.473 *Byte*. kapasitas yang ditunjukkan wireshark belum dinormalisasi sehingga hasil yang ditampilkan relatif berbeda dengan kapasitas *file* semula.

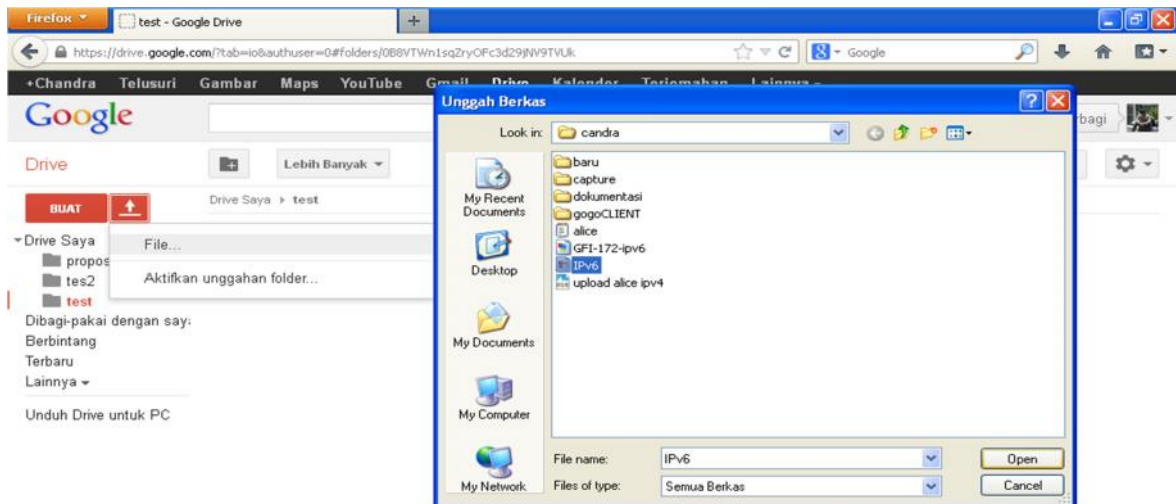
7. Lakukan kembali langkah 1 sampai 6 dengan menggunakan *interface* gogo6.



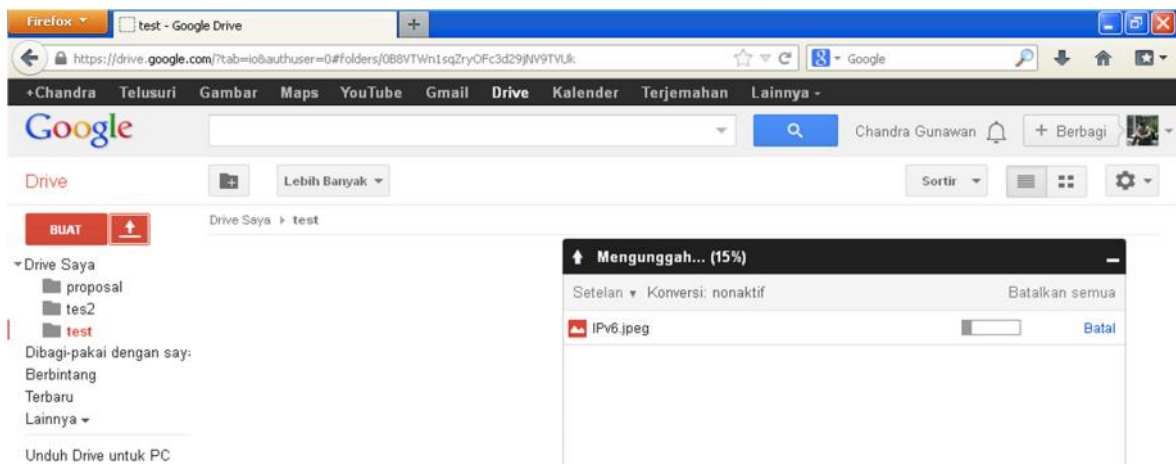
Gambar 4.26. Prosedur *uploading* dengan *google drive*



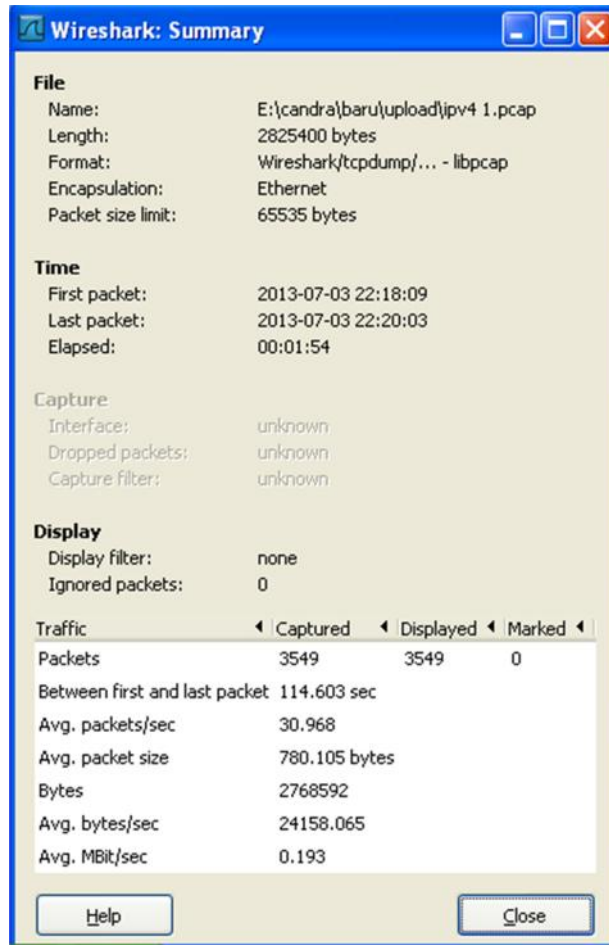
Gambar 4.27. Pemilihan *interface* untuk proses *capture*



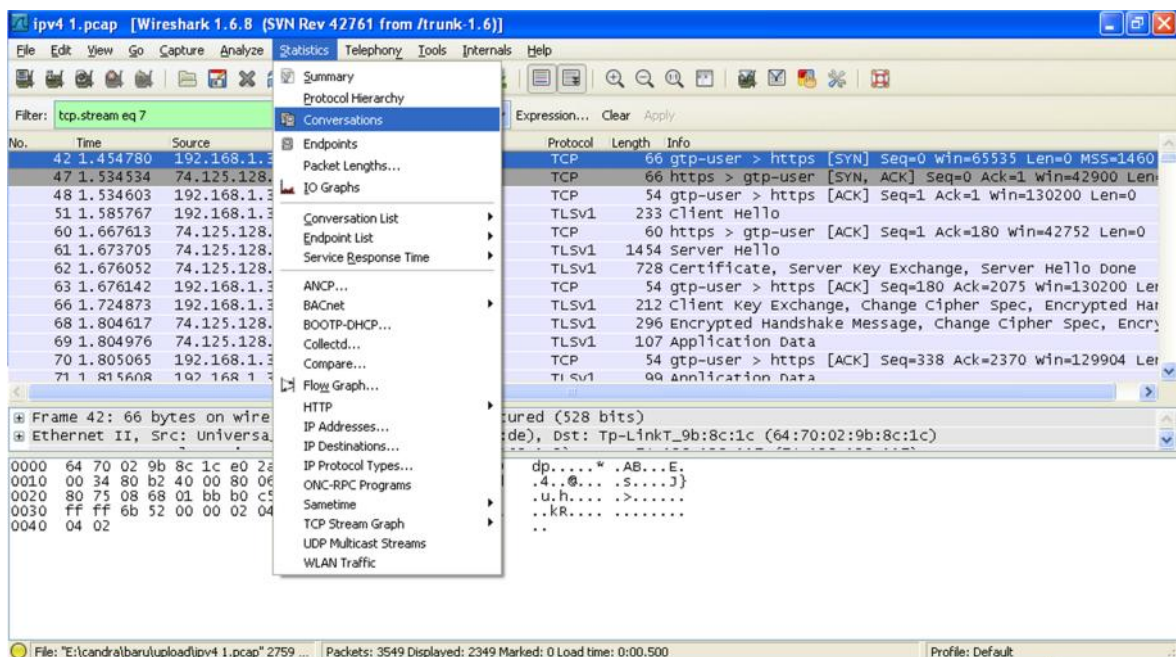
Gambar 4.28. Memilih *file* yang akan di-upload



Gambar 4.29. Proses *upload*



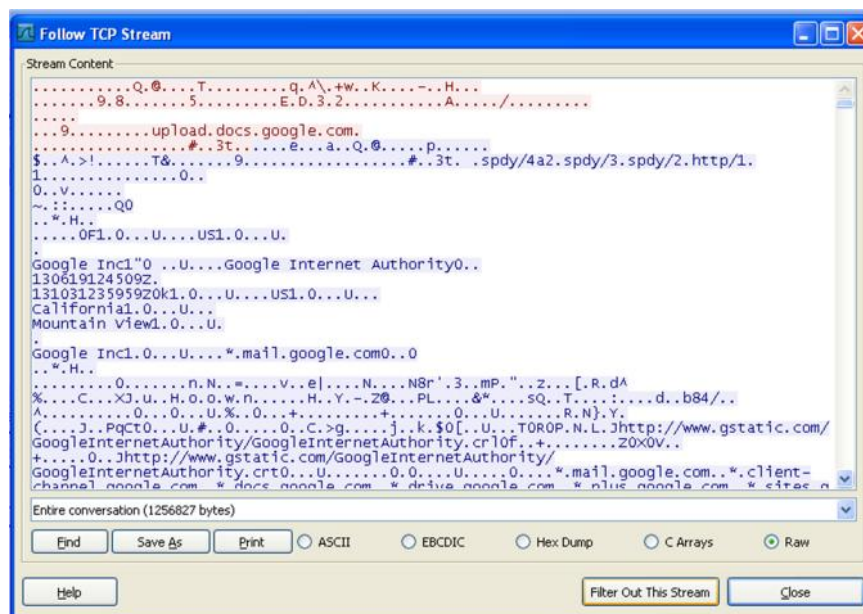
Gambar 4.30. Hasil *capturing upload* pada wireshark



Gambar 4.31. Menu menuju *Conversations*

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A->B	Bytes A->B	Packets B->A	Bytes B->A	Rel Start	Duration	bps A->B	bps B->A
192.168.1.3	veritas-ud	173.194.38.169	https	14	1 061	7	520	7	541	0.000000000	59.1012	70.39	73.23
192.168.1.3	lv-jc	74.125.135.189	https	37	10 931	15	6 366	22	4 565	0.024844000	104.2629	488.46	350.27
192.168.1.3	avenue	173.194.38.160	https	40	11 254	20	7 605	20	3 649	0.737671000	113.8655	534.31	256.37
192.168.1.3	docent	173.194.38.160	https	40	8 534	20	4 115	20	4 419	1.385978000	102.9084	319.90	343.53
192.168.1.3	gtp-user	74.125.128.117	https	2 349	2 014 201	1 375	1 947 280	974	66 921	1.454780000	102.5296	151938.89	5221.59
192.168.1.3	dynamic3d	173.194.38.174	https	16	1 175	8	574	8	601	11.969078000	59.1275	77.66	81.32
192.168.1.3	gris	74.125.135.84	https	12	848	5	338	7	510	21.174587000	0.0554	48778.73	73601.04
192.168.1.3	disrpn	118.98.42.89	http	737	645 378	305	18 256	432	627 122	37.366459000	68.4006	2135.19	73346.94
192.168.1.3	unbind-cluster	173.194.38.163	https	12	879	6	429	6	450	42.969534000	0.0992	34584.92	36277.89
192.168.1.3	zmyed-zpp	173.194.38.170	https	13	939	6	429	7	510	47.969567000	0.2844	12069.29	14348.11
192.168.1.3	appworxsr	173.194.38.175	https	11	819	6	429	5	390	47.969782000	0.2935	11692.00	10629.09
192.168.1.3	tdmop	173.194.38.172	https	13	939	6	429	7	510	49.969578000	0.0927	37017.46	44006.77
192.168.1.3	connect	74.125.235.30	https	12	879	6	429	6	450	49.969718000	0.1002	34264.83	35942.13
192.168.1.3	las-reg	173.194.38.173	https	13	939	6	429	7	510	52.969760000	0.0921	37278.42	44317.00
192.168.1.3	lv-ffx	173.194.38.186	https	10	759	6	429	4	330	53.969708000	0.1270	27016.60	20782.00
192.168.1.3	aura	173.194.38.178	https	12	941	7	551	5	390	53.969845000	0.7610	5792.09	4099.66
192.168.1.3	lv-auth	173.194.38.162	https	13	939	6	429	7	510	55.969741000	0.1040	32988.26	39216.82

Gambar 4.32. Conversations Wireshark



Gambar 4.33. Follow TCP Stream

Pada gambar 4.30 angka 3549 merupakan jumlah paket TCP yang terdiri dari *send* dan *recieve* yang jumlah paket *recieve* nya adalah 1375 paket, kemudian angka 102,5296 adalah *delay* paket TCP tersebut. Dari beberapa nilai ini, dapat dihitung persentasi paket *sent* dengan rumus:

$$\text{Persentase Paket Sent} = \frac{\text{jumlah Paket Download}}{\text{jumlah Paket yang Tersaring}} \times 100\%$$

$$\begin{aligned} \text{Persentase Paket Sent} &= \frac{1375}{2349} \times 100\% \\ &= 58,54 \% \end{aligned}$$

Jadi, persentase paket *sent* yang didapat dari rumus tersebut adalah 58,54 %. Maka untuk menentukan *delay upload* murni dari suatu percobaan *upload* dapat dilihat dengan menggunakan rumus seperti berikut.

$$\text{Delay Upload} = \text{Delay yang tersaring} \times \text{Persentase Paket Download}$$

$$\text{Delay Upload} = 102,5296 \text{ second} \times 56,21\%$$

$$= \mathbf{60,016262 \text{ second}}$$

Jadi, delay murni pada sebuah proses *upload* untuk percobaan pertama ialah 60,016262 *second*. Nilai *delay* tidak *absolute* karena dipengaruhi banyak faktor dan kondisi jaringan pada saat pengiriman paket data.

Selanjutnya, persentase paket pelengkap adalah persentase paket-paket yang bukan berisi data transfer namun banyaknya jumlah paket ini menimbulkan *delay* yang besar. Paket ICMP merupakan salah satu paket pelengkap dalam hal ini, maka untuk mengetahui persentasenya digunakan rumus sebagai berikut:

$$\text{Persentase Paket Pelengkap} = \frac{\text{jumlah Paket Total} - \text{Jumlah Paket Tersaring}}{\text{Jumlah Paket Total}} \times 100\%$$

$$\text{Persentase Paket Pelengkap} = \frac{3549 - 2349}{3549} \times 100\%$$

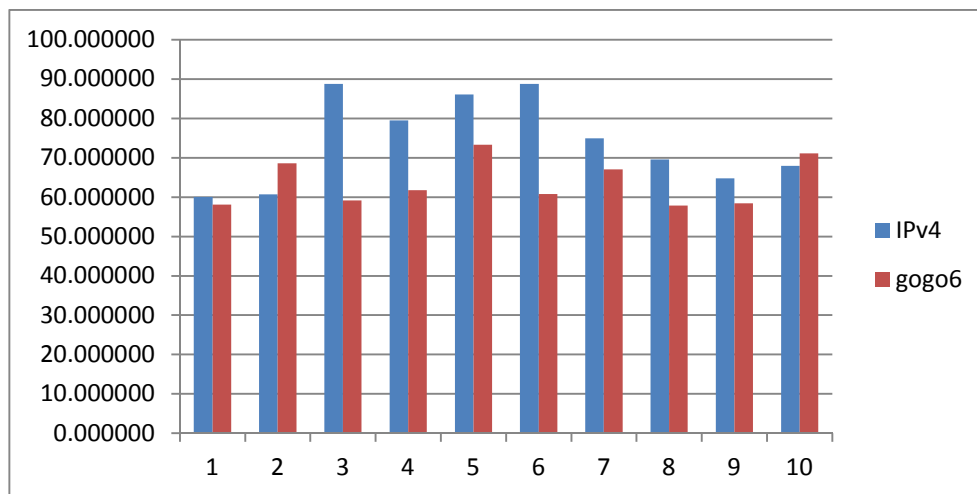
$$= \mathbf{33,81 \%}$$

Berdasarkan rumus di atas, penulis melakukan percobaan sebanyak 10 kali untuk mendapatkan persentase paket sent rata-rata, persentase paket pelengkap dan *delay upload* rata-rata yang akurat karena jika hanya satu kali percobaan saja belum bisa membuktikan kualitas gogo6 dengan IPv4 secara keseluruhan hasil percobaan dapat dilihat pada tabel 4.2.

Tabel 4.2. Data *upload* dengan wireshark

Interface	No.	Jumlah Paket Total (Packets)	Jumlah Paket yang Tersaring (Packets)	Persentase Paket Pelengkap (%)	Delay yang Disaring (Seconds)	Jumlah Recieve (Packets)	Jumlah Paket Sent (Packets)	Persentase Paket Upload (%)	Delay Upload (Seconds)
IPv4	1	3549	2349	33,81	102,5296	974	1375	58,54	60,016262
	2	4145	2297	44,58	104,4711	962	1335	58,12	60,717857
	3	3533	2355	33,34	149,9106	961	1394	59,19	88,736890
	4	3434	2336	31,97	135,0718	961	1375	58,86	79,505019
	5	3452	2322	32,73	147,9093	971	1351	58,18	86,057478
	6	3575	2390	33,15	149,2796	969	1421	59,46	88,755779
	7	3470	2322	33,08	127,4017	965	1366	58,83	74,948631

	8	3413	2344	31,32	118,7181	970	1374	58,62	69,589876
	9	3418	2308	32,48	111,4778	967	1341	58,10	64,771113
	10	3413	2312	32,26	116,0165	958	1354	58,56	67,943919
gogo6	1	2642	2471	6,47	103,4700	1084	1387	56,13	58,078871
	2	2764	2595	6,11	120,8039	1121	1474	56,80	68,618477
	3	2623	2490	5,07	105,9646	1101	1389	55,78	59,110373
	4	2639	2483	5,91	109,9999	1090	1393	56,10	61,711583
	5	2785	2478	11,02	131,7681	1100	1378	55,61	73,275400
	6	2575	2443	5,13	108,2975	1072	1371	56,12	60,776043
	7	2688	2548	5,21	120,4306	1130	1418	55,65	67,021425
	8	2548	2404	5,65	104,2028	1070	1334	55,49	57,823018
	9	2553	2436	4,58	104,3017	1072	1364	55,99	58,402101
	10	2749	2609	5,09	127,8082	1158	1451	55,62	71,080758



Gambar 4.34. Grafik *delay* tiap percobaan *upload*

Dilakukan 10 percobaan untuk menemukan hasil yang akurat pada penelitian ini. Balok diagram berwarna biru menggambarkan *delay upload* rata-rata dari IPv4 dan balok merah adalah *delay upload* rata-rata gogo6. Dari 10 percobaan diambil 1 *delay* rata-rata dengan menggunakan rumus berikut:

$$\text{Delay upload rata-rata pada IPv4} = \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n}$$

$$\begin{aligned} \text{Delay upload rata-rata pada IPv4} &= \frac{60,016262+60,717857+88,736890+79,505019+86,057478+88,755779+74,948631+69,589876+64,771113+67,943919}{10} \\ &= \mathbf{74,104283 \text{ seconds}} \end{aligned}$$

$$\text{Delay upload rata-rata pada gogo6} = \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n}$$

$$\begin{aligned} \text{Delay upload rata-rata pada gogo6} &= \frac{58,078871+68,618477+59,110373+61,711583+73,275400+60,776043+67,021425+57,823018+58,402101+71,080758}{10} \end{aligned}$$

$$= 63,589805 \text{ seconds}$$

Jadi, *delay upload* rata-rata keseluruhan pada IPv4 adalah 74,104283 *seconds* dan *delay upload* rata-rata keseluruhan *tunnel broker* gogo6 adalah 63,589805 *seconds*.

Jika diamati lebih jauh, ada perbedaan signifikan dari jumlah paket *recieve* antara percobaan *upload* IPv4 dengan gogo6 pada tabel 4.2. untuk menentukan rata-rata jumlah paket *recieve* maka digunakan rumus sebagai berikut :

$$\text{Jumlah paket } \textit{recieve} \text{ rata-rata IPv4} = \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n}$$

$$\begin{aligned} \text{Jumlah paket } \textit{recieve} \text{ rata-rata IPv4} &= \frac{974+962+961+961+971+969+962+970+967+958}{10} \\ &= 966 \text{ packets} \end{aligned}$$

$$\text{Jumlah paket } \textit{recieve} \text{ rata-rata gogo6} = \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n}$$

$$\begin{aligned} \text{Jumlah paket } \textit{recieve} \text{ rata-rata gogo6} &= \frac{1084+1121+1001+1090+1100+1072+1130+1070+1072+1158}{10} \\ &= 1100 \text{ packets} \end{aligned}$$

Jadi, jumlah paket *recieve* rata-rata pada proses *upload* keseluruhan pada IPv4 adalah 966 *packets* dan jumlah paket *recieve* rata-rata pada proses *upload* keseluruhan *tunnel broker* gogo6 adalah 1100 *packets*. Hal ini mengindikasikan bahwa jumlah paket *recieve* pada gogo6 lebih banyak dibanding jumlah paket pada IPv4 sehingga waktu yang dibutuhkan gogo6 untuk proses *upload* relatif lebih cepat dibanding waktu yang dibutuhkan IPv4.

Selain itu, ada perbedaan signifikan dari jumlah paket total antara percobaan *upload* IPv4 dengan gogo6 pada tabel 4.2. untuk menentukan rata-rata jumlah paket *recieve* maka digunakan rumus sebagai berikut :

$$\text{Persentase Paket Pelengkap rata-rata IPv4} = \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n}$$

$$\begin{aligned} \text{Persentase Paket Pelengkap rata-rata IPv4} &= \frac{33,81+44,58+33,34+31,97+32,73+33,15+33,08+31,32+32,48+32,26}{10} \\ &= 33,87 \% \end{aligned}$$

$$\text{Persentase Paket Pelengkap rata-rata gogo6} = \frac{P1+P2+P3+P4+P5+P6+P7+P8+P9+P10}{n}$$

$$\begin{aligned} \text{Persentase Paket Pelengkap rata-rata gogo6} &= \frac{6,47+6,11+5,07+5,91+11,02+5,13+5,21+5,65+4,58+5,09}{10} \\ &= 6,03 \% \end{aligned}$$

Jadi, persentase paket pelengkap rata-rata pada proses *upload* keseluruhan pada IPv4 adalah 33,87 % dan jumlah paket *recieve* rata-rata pada proses *upload* keseluruhan *tunnel broker* gogo6 adalah 6,03 %

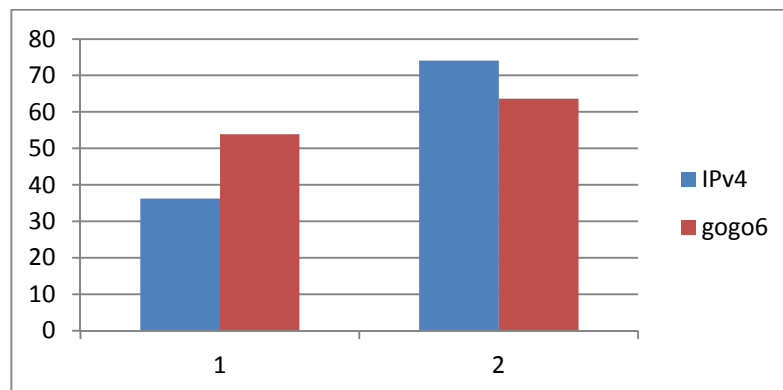
Hal ini mengindikasikan bahwa jumlah paket *recieve* pada gogo6 lebih banyak dibanding jumlah paket pada IPv4 sehingga waktu yang dibutuhkan gogo6 untuk proses *upload* relatif lebih cepat dibanding waktu yang dibutuhkan IPv4. Selain persentase paket pelengkap pada IPv4 lebih besar yaitu 33,87% dibanding persentase paket pelengkap pada gogo6 dengan nilai 6,03%.

4.2.3. Delay Keseluruhan

Dari nilai yang dianalisis baik nilai *download* dan *upload* maka didapat *delay* rata-rata pada tabel 4.3 sebagai berikut.

Tabel 4.3. Perbandingan *delay*

Aktifitas	nomor	Interface	Delay Murni
<i>download</i>	1	IPv4	36,255784
	2	gogo6	53,897352
<i>upload</i>	1	IPv4	74,104283
	2	gogo6	63,589805



Gambar 4.35. Grafik perbandingan *delay* rata-rata semua percobaan

Untuk mengetahui seberapa besar perbandingan *delay* rata-rata antara IPv4 dengan *tunnel broker* gogo6, maka diperoleh hasil sebagai berikut:

Perbandingan *download*:

$$\text{Gogo6} : \text{IPv4} = \frac{\text{gogo6}}{\text{IPv4}} : \frac{\text{IPv4}}{\text{IPv4}}$$

$$\begin{aligned}
&= \frac{53,897352}{36,255784} : \frac{36,255784}{36,255784} \\
&= 1,4865863:1 \text{ (1,4:1)}
\end{aligned}$$

Perbandingan *upload*:

$$\begin{aligned}
\text{Gogo6} : \text{IPv4} &= \frac{\text{gogo6}}{\text{IPv4}} : \frac{\text{IPv4}}{\text{IPv4}} \\
&= \frac{63,589805}{74,104283} : \frac{74,104283}{74,104283} \\
&= 0,8581124:1 \text{ (0,8:1)}
\end{aligned}$$

Hasil ini mendeskripsikan bahwa waktu yang dibutuhkan saat aktifitas *download* pada gogo6 **1,4** kali lebih lama dibanding waktu yang dibutuhkan IPv4 untuk data yang sama. Pada aktifitas *upload* gogo6 dan IPv4 memiliki perbandingan sebesar **0,8:1**.

4.2.4. Analisis Hasil Implementasi

Dalam implementasi interkoneksi IPv6 dan IPv4 dengan Aplikasi *tunnel broker* di laboratorium jaringan terdapat beberapa analisis sebagai berikut :

1. Gogo6 pada implementasinya terdapat dua tahapan transisi yaitu enkapsulasi dari IPv6 ke IPv4 yang berjalan pada *gogoCLIENT* dan dekapsulasi dari IPv4 ke IPv6 yang berjalan pada *gogoSERVER*. Hal ini yang menyebabkan pada beberapa percobaan mengalami delay yang relatif besar.
2. Kapasitas *file* yang di-*capture* oleh wireshark menunjukkan perbedaan dengan kapasitas *file* yang sebenarnya, ini disebabkan oleh hasil *capturing* masih dalam bentuk yang bersegmen-segmen sehingga *file* ini harus dinormalisasikan kembali agar *file* menjadi utuh seperti semula.
3. Pada proses *upload* hasil analisis menunjukkan *delay* murni IPv4 cenderung lebih besar dibanding gogo6, ini disebabkan persentase paket pelengkap pada IPv4 dengan rata-rata 33,87 % lebih besar dibandingkan persentase paket pelengkap pada gogo6 dengan rata-rata 6,03 %. Hal ini menunjukkan bahwa untuk aktifitas *Upload tunnel broker* lebih unggul dibanding IPv4.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

1. *Tunnel broker* dapat mengatasi permasalahan koneksi IPv6 dengan menggunakan perangkat dan infrastruktur IPv4. Untuk aktifitas interkoneksi IPv6, *tunnel broker* dapat menjadi suatu solusi dan dapat diimplementasikan pada jaringan dengan beberapa komputer seperti pada Laboratorium Komputer.
2. Dari hasil percobaan pada *delay* rata-rata ketika dilakukan proses transfer diperoleh *delay* rata-rata *download* IPv4 sebesar 36,255784 *second* dan *delay* rata-rata *download tunnel broker* sebesar 53,897352 *second*. Untuk *upload*, *delay* rata-rata IPv4 sebesar 74,104283 *second* dan *delay* rata-rata *upload tunnel broker* 63,589805 *second*. Ini menunjukkan bahwa IPv6 lebih cepat dalam proses *upload* dibanding IPv4 karena paket pelengkap yang bukan termasuk data yang ditransfer pada IPv6 lebih sedikit dibanding IPv4 sehingga waktu yang diproses pun lebih sedikit.

5.2. Saran

1. Penelitian yang berkelanjutan dengan modal dasar ini perlu terus dikembangkan antara lain, untuk mekanisme transisi dapat menggunakan *DualStack* baik secara simulasi maupun implementasi, untuk *tunneling* bisa diambil permasalahan tentang keamanan transisi *tunneling*.
2. Laboratorium komputer tidak memberikan akses internet kepada mahasiswa untuk fasilitas pembelajaran sehingga perlu diperhatikan oleh pihak UIN SUSKA agar mahasiswa dapat menggunakan fasilitas internet untuk mendukung implementasi mekanisme transisi *tunneling* ini yang diaplikasikan dengan *tunnel broker*.

DAFTAR PUSTAKA

- Alan Sohn: *IPv6*. <http://en.wikipedia.org/wiki/IPv6>. 20 November 2011.
- Syarif, Mochammad Averoes: *Analisis Unjuk kerja Aplikasi Video Streaming pada Jaringan IPv6 dan IPv6-Dual Stack dengan menggunakan PC Router & Emulator GNS3*. Universitas Indonesia; desember 2012.
- Artondo, Reko: *Analisa dan Implementasi IPv6 Tunnel Broker untuk Interkoneksi antara IPv6 dan IPv4*. Universitas Diponegoro; 2010.
- Indah, Renny lestari: *Menganalisa Kinerja Antara Metode Tunneling 6to4 dengan Metode Dual Stack Berbasis Protokol IPv6 Menggunakan Router Mikrotik (Studi Kasus PT. Time Excelindo)*. STMIK Amikom Yogyakarta; 2011.
- Deering, S: *RFC 1883: Internet Protocol, Version 6 (IPv6) Specification*, Network Working Group, URL: <http://www.cis.ohio-state.edu/htbin/rfc/>. 1995.
- Anonymous: *Conten Gogo6 Guide*, http://content.gogo6.com/gogo_dc_0010_gogoclient_guide.pdf; 2009. (Diakses: 26 Mei 2013)
- Bahrul Ulum: *TCP dan UDP*, <http://www.bahrul-ulum.com/tcp-dan-udp>, 2012. Diakses tanggal 26 Mei 2013.
- Erik Rodriguez: *TCP and UDP*, <http://www.skullbox.net/tcpudp.php>, Orlando Tech Works, LLC, 2012. (Diakses: 26 Mei 2013)
- Anonymous; *TCP dan UDP*, <http://thelolbee.wordpress.com/2010/05/20/tcp-dan-udp/>. 20 Mei 2010. (Diakses: 26 Mei 2013)
- Eko, Mohamad Ari Bowo: *Penggunaan IPv6 Sebagai Solusi Pengganti IPv4 dalam Penanganan Keterbatasan IP Address di Jaringan Internet Masa Depan*, Universitas Sebelas Maret; Januari 2009.
- Tanenbaum Andrew.S: *Jaringan komputer Jilid 1*, Pearson Education Asia, 1998.
- Stallings, William, diterjemahkan Abdul Hafedh Al-Hamdany, B.Sc.,M.Sc: *Jaringan Komputer*. Salemba teknika, Pearson Education Asia Pte.Ltd, Jilid 2. 2002.
- Shalini, D. Punithavathani, K.Sankaranarayanan: *IPv4/IPv6 Transition Mechanisms*, European Journal of Scientific Research, pp.110-124: 2009.
- Anonymous: *Apa Itu IPv6, 10 Hal Yang Perlu Diketahui & Artinya Bagi Kita ?*, <http://www.iniunik.web.id/2011/06/apa-itu-ipv6-10-hal-yang-perlu.html#ixzz2DQii2IU>. 2011. (Diakses: 27 November 2012)
- Anonymous: *Alamat IP versi 4*. http://id.wikipedia.org/wiki/Alamat_IP_versi_4. (Diakses: 30 Maret 2013)
- Anonymous: *Tunnel Broker*. http://en.wikipedia.org/wiki/Tunnel_broker. (Diakses: 30 Maret 2013)

Anonymous: *Internet Control Message Protocol*. http://id.wikipedia.org/wiki/Internet_Control_Message_Protocol. (Diakses: 30 Maret 2013)

Dwi, Endi Kristianto: *Menghitung Delay Paket Pada Jaringan Menggunakan Wireshark*. <http://ilmukomputer.com>, 2013. (Diakses: 30 Maret 2013)